Dell PowerVault MD 32XX/34XX/36XX/38XX Series Storage Arrays
CLI Guide



Notes, Cautions, and Warnings

NOTE: A NOTE indicates important information that helps you make better use of your computer.

CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

Copyright © 2015 Dell Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Dell™ and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

2015 - 02

Rev. A01

Contents

1 CLI Command Updates....16
- New Commands....16
- Updated Commands....16

2 About The Command Line Interface....18
- How To Use The Command Line Interface....19
  - Usage Notes....19
  - CLI Commands....19
  - Command Line Parameters....21
- Formatting Considerations....25
- Detailed Error Reporting....25
- Exit Status....26
- Usage Examples....27

3 About The Script Commands....29
- Script Command Structure....30
- Script Command Synopsis....31
- Recurring Syntax Elements....33
- Usage Guidelines....38
- Adding Comments To A Script File....38

4 Configuring A Storage Array....39
- Configuring A Storage Array....40
  - Determining What Is On Your Storage Array....40
  - Saving A Configuration To A File....42
  - Using The Create Virtual Disk Command....42
  - Using The Auto Configure Command....45
- Modifying Your Configuration....47
  - Setting The Storage Array Password....47
  - Setting Up SMTP And SNMP Alerts....48
  - Setting The RAID Controller Module Clocks....48
  - Setting The Storage Array Host Type....48
  - Setting Modification Priority....49
  - Assigning Global Hot Spares....50
  - Selecting The Event Levels For Alert Notifications....50
  - Configuring Alert Notifications....51

5 Using The Snapshot Feature....52

Before Using Snapshot CLI Commands....52
Snapshot Images And Groups....52
Snapshot Groups And Snapshot Consistency Groups....53
Snapshot Groups....53
Snapshot Consistency Groups....53
Understanding Snapshot Repositories....54
Snapshot Consistency Group Repositories....54
Consistency Groups And Remote Replication....54
Creating Snapshot Images....55
Creating A Snapshot Image....55
Deleting A Snapshot Image....55
Creating A Consistency Group Snapshot Virtual Disk....56
Creating A Consistency Group Snapshot Image....57
Deleting A Snapshot Virtual Disk Or A Consistency Group Snapshot Virtual Disk....57
Deleting A Consistency Group Snapshot Image....57
Scheduling Snapshot Images....58
Starting, Stopping And Resuming A Snapshot Rollback....58
Creating A Snapshot Group....59
Deleting A Snapshot Group....60
Reviving A Snapshot Group....60
Creating A Consistency Group....60
Deleting A Consistency Group....61
Setting Consistency Group Attributes....61
Adding A Member Virtual Disk To A Consistency Group....62
Removing A Member Virtual Disk From A Consistency Group....62
Changing The Pre-read Consistency Check Setting Of An Overall Repository Virtual Disk....63
Setting Snapshot Virtual Disk Repository Virtual Disk Capacity....64
Setting Snapshot Group Repository Virtual Disk Capacity....65
Reviving Disk Groups, Physical Disks, Snapshot Groups, And Snapshot Virtual Disks....65

6 Using The Snapshot (Legacy) Feature....67

Using Host Servers To Create An Initial Snapshot Virtual Disk....68
Creating A Snapshot Virtual Disk....69
Enabling The Snapshot Virtual Disk Feature....69
Creating A Snapshot Virtual Disk With User-Assigned Physical Disks....69
Preparing Host Servers To Create An Initial Snapshot Virtual Disk....70
Creating The Initial Snapshot Virtual Disk....70
Creating A Snapshot Virtual Disk With Software-Assigned Physical Disks....71
Creating A Snapshot Virtual Disk By Specifying A Number Of Physical Disks....72
User-Defined Parameters....73
Names Of Snapshot Virtual Disks And Repository Virtual Disks....75
Changing Snapshot Virtual Disk Settings....75

Stopping And Deleting A Snapshot Virtual Disk....76
Re-creating The Snapshot Virtual Disk....76
Preparing Host Servers To Re-create A Snapshot Virtual Disk....77
Re-creating A Snapshot Virtual Disk....77

7 Using The Virtual Disk Copy Feature....78
About Virtual Disk Copy....78
Virtual Disk Copy Types....78
Creating A Virtual Disk Copy....80
Enabling The Virtual Disk Copy Feature....80
Determining Virtual Disk Copy Candidates....80
Creating A Virtual Disk Copy....80
Preparing Host Servers To Create A Virtual Disk Copy....81
Copying The Virtual Disk....81
Viewing Virtual Disk Copy Properties....82
Changing Virtual Disk Copy Settings....83
Recopying A Virtual Disk....84
Preparing Host Servers To Recopy A Virtual Disk....84
Recopying The Virtual Disk....85
Stopping A Virtual Disk Copy....85
Removing Copy Pairs....86
Interaction With Other Features....86
About Snapshot (Legacy) Premium Features With Virtual Disk Copy....86

8 Using The Remote Replication Premium Feature....87
How Remote Replication Works....87
Replication Pairs And Replication Repositories....88
Differences Between Remote Replication And Remote Replication (Legacy) Features....88
Link Interruptions Or Secondary Virtual Disk Errors....88
Resynchronization....89
Remote Replication Group....90
Previous Users Of Remote Replication (Legacy) Premium Feature....90
Remote Replication Requirements And Restrictions....90
Primary And Secondary Virtual Disks....91
Setting Up Remote Replication....91
Enabling The Remote Replication Premium Feature....91
Activating The Remote Replication Premium Feature....91
Creating A Remote Replication Group....91
Adding Primary Virtual Disk To Remote Replication Group....92
Changing Remote Replication Group Settings....92
Adding Secondary Virtual Disk To Remote Replication Group....93
Suspending A Remote Replication Group....93

Deleting A Remote Replication Group....94
Removing A Virtual Disk Or Repository Virtual Disk From A Remote Replication Group....94
Disabling The Remote Replication Premium Feature....95
Deactivating The Remote Replication Premium Feature....95
Interaction With Other Premium Features....95
Standard Remote Replication Commands....95

9 Using The Remote Replication (Legacy) Premium Feature....97
How Remote Replication (Legacy) Works....97
Replication Repository Virtual Disks....98
Replication Relationships....98
Data Replication....99
Link Interruptions Or Secondary Virtual Disk Errors....100
Resynchronization....100
Creating A Remote Replicated Pair....101
Performance Considerations....101
Enabling The Remote Replication (Legacy) Premium Feature....102
Activating The Remote Replication (Legacy) Premium Feature....102
Determining Candidates For A Remote Replicated Pair....104
Creating A Remote Replicated Pair....104
Changing Remote Replication (Legacy) Settings....105
Suspending And Resuming A Replication Relationship....106
Removing A Replication Relationship....107
Deleting A Primary Virtual Disk Or A Secondary Virtual Disk....107
Disabling The Remote Replication (Legacy) Premium Feature....108
Deactivating The Remote Replication (Legacy) Premium Feature....108
Interaction With Other Premium Features....108
Snapshot Virtual Disks....108
Virtual Disk Copy....109
Storage Partitioning....109
Virtual Disk Expansion....110

10 Maintaining A Storage Array....111
Routine Maintenance....111
Running A Media Scan....111
Running A Consistency Check With A Media Scan....112
Resetting A RAID Controller Module....113
Enabling RAID Controller Module Data Transfer....113
Resetting Battery Age....113
Removing Persistent Reservations....113
Synchronizing RAID Controller Module Clocks....114
Locating Physical Disks....114

Performance Tuning....114
Monitoring Performance....114
Changing RAID Levels....115
Changing Segment Size....115
Defragmenting A Disk Group....116
Using SSD Cache....116
Enabling SSD Cache For A Virtual Disk....116
Creating SSD Cache....117
Adding Physical Disks To Existing SSD Cache....117
Removing Physical Disks From Existing SSD Cache....117
SSD Cache Performance Modeling....117
Additional Information About SSD Cache And Performance Modeling....118
Troubleshooting and Diagnostics....118
Collecting Physical Disk Data....118
Diagnosing A RAID Controller Module....118
Recovery Operations....119
Setting RAID Controller Module Operational Mode....119
Changing RAID Controller Module Ownership....120
Initializing A Physical Disk....120
Reconstructing A Physical Disk....121
Initializing A Virtual Disk....121
Redistributing Virtual Disks....121

11 Script Commands....122

Command Formatting Rules....122
Firmware Compatibility Levels....124
Commands Listed By Function....124
Disk Group Commands....124
Enclosure Commands....125
Host Topology Commands....125
iSCSI Commands....126
Physical Disk Commands....126
RAID Controller Module Commands....127
Remote Replication Commands....128
Session Command....129
Show String Command....129
Snapshot Commands....129
SSD Cache Commands....129
Storage Array Commands....130
Virtual Disk Commands....132
Virtual Disk Copy Commands....132

12 Commands Listed Alphabetically..... 133
Activate Remote Replication Feature..... 133
Activate Storage Array Firmware..... 136
Add Member To Consistency Group..... 136
Add Physical Disks To SSD Cache..... 138
Add Virtual Disk To Remote Replication Group..... 139
Autoconfigure Storage Array..... 141
Autoconfigure Storage Array Hot Spares..... 143
Blink SSD Cache..... 143
Cancel Remote Replication Group Role Reversal..... 144
Change SSD Cache Application Type..... 144
Check Remote Replication Status..... 145
Check Remote Replication Group Consistency..... 145
Check Repository Consistency..... 146
Check Storage Array Connectivity..... 147
Check Virtual Disk Consistency..... 148
Clear Physical Disk Port Statistics..... 149
Clear Remote Replication Fault..... 150
Clear Storage Array Configuration..... 151
Set Storage Array Core Dump..... 152
Clear Storage Array Event Log..... 152
Clear Storage Array Firmware Pending Area..... 153
Clear Storage Array Recovery Mode..... 153
Clear Virtual Disk Reservations..... 153
Clear Virtual Disk Unreadable Sectors..... 154
Configure Automatic Support Bundle Collection..... 155
Convert Snapshot (Legacy) Virtual Disk To Snapshot Group..... 155
Create Consistency Group..... 156
Create Consistency Group Snapshot Image..... 158
Create Consistency Group Snapshot Virtual Disk..... 159
Create Consistency Group Snapshot Virtual Disk Mapping..... 161
Create Disk Group..... 162
Create Disk Pool..... 165
Create Host..... 169
Create Host Group..... 169
Create Host Port..... 170
Create iSCSI Initiator..... 171
Create RAID Virtual Disk (Automatic Physical Disk Select)..... 171
Create RAID Virtual Disk (Free Extent Base Select)..... 174
Create RAID Virtual Disk (Manual Physical Disk Select)..... 176
Create RAID Virtual Disk On A Disk Pool..... 179

Create Read-Only Snapshot Virtual Disk....181
Create Remote Replication....182
Create Remote Replication Group....184
Create Snapshot Group....187
Create Snapshot Image....189
Create Snapshot (Legacy) Virtual Disk....190
Create Snapshot Virtual Disk....196
Create SSD Cache....198
Create Storage Array Security Key....199
Create Virtual Disk Copy....201
Deactivate Remote Replication (Legacy)....203
Delete Consistency Group....203
Delete Consistency Group Snapshot Image....204
Delete Consistency Group Snapshot Virtual Disk....205
Delete Disk Group....205
Delete Disk Pool....206
Delete Host....206
Delete Host Group....207
Delete Host Port....207
Delete iSCSI Initiator....208
Delete Snapshot Group....209
Delete Snapshot Image....209
Delete Snapshot (Legacy) Virtual Disk....211
Delete Snapshot Virtual Disk....212
Delete Snapshot Virtual Disk or Delete Snapshot Virtual Disk Schedule....212
Delete SSD Cache....213
Delete Virtual Disk....213
Delete Virtual Disk On A Disk Pool....214
Diagnose RAID Controller Module....215
Diagnose RAID Controller Module iSCSI Host Cable....217
Diagnose Remote Replication....218
Disable Storage Array Feature....219
Display Automatic Support Bundle Collection Configuration....220
Display Support Bundle Collection Schedule....221
Download Enclosure Management Module Firmware....221
Download Physical Disk Firmware....222
Download Storage Array Firmware/NVSRAM....223
Download Storage Array Physical Disk Firmware....223
Download Storage Array NVSRAM....224
Enable Disk Group Security....225
Enable Disk Pool Security....226
Enable or Disable ODX....226

Enable Or Disable SSD Cache For A Virtual Disk 227
Enable or Disable VAAI 228
Enable RAID Controller Module Data Transfer 228
Enable Storage Array Feature 229
Establish Remote Replicated Pair 229
Increase Virtual Disk Capacity On A Disk Pool 230
Load Storage Array DBM Database 231
Modify Disk Pool 233
Recopy Virtual Disk Copy 233
Recover RAID Virtual Disk 235
Re-Create Remote Replication Repository Virtual Disk 239
Re-Create Snapshot 242
Reduce Disk Pool Capacity 244
Register SNMP Community 245
Register SNMP Trap Destination 245
Reinitialize Thin Virtual Disk 246
Remove Incomplete Remote Replicated Pair From Remote Replication Group 247
Remove Member Virtual Disk From Consistency Group 248
Remove Physical Disks From SSD Cache 248
Remove Remote Replication 249
Remove Virtual Disk Copy 250
Remove Virtual Disk From Remote Replication Group 250
Remove Virtual Disk LUN Mapping 251
Rename A Snapshot Virtual Disk 252
Rename SSD Cache 253
Repair Virtual Disk Consistency 253
Replace Physical Disk 254
Reset iSCSI IP Address 255
Reset Physical Disk 256
Reset RAID Controller Module 257
Reset Remote Replication Group Statistics 258
Reset Storage Array Battery Install Date 259
Reset Storage Array iSCSI Baseline 260
Reset Storage Array SAS PHY Baseline 260
Reset Storage Array Virtual Disk Distribution 261
Resume Consistency Group Snapshot Virtual Disk 261
Resume Remote Replication 262
Resume Remote Replication Group 263
Resume Snapshot Image Rollback 264
Resume Snapshot (Legacy) Rollback 265
Resume Snapshot Virtual Disk 265
Resume SSD Cache 267

Revive Disk Group....267
Revive Disk Pool....268
Revive Physical Disk....268
Revive Snapshot Group....269
Revive Snapshot Virtual Disk....269
Save Enclosure Log Data....270
Save Input Output RAID Controller Module (IOC) Dump....270
Save Physical Disk Log....271
Save RAID Controller Module NVSRAM....271
Save Remote Replication Group Statistics....272
Save Storage Array Configuration....274
Save Storage Array Core Dump....275
Save Storage Array DBM Database....275
Save Storage Array DBM Validator Information File....277
Save Storage Array Events....278
Save Storage Array iSCSI Statistics....279
Save Storage Array Performance Statistics....280
Save Storage Array SAS PHY Counts....280
Save Storage Array State Capture....281
Save Storage Array Support Data....281
Schedule Support Bundle Collection Configuration....282
Scheduling Snapshots....283
Set Consistency Group Attributes....285
Set Consistency Group Snapshot Virtual Disk....286
Set Disk Group....286
Set Disk Group Forced State....288
Set Disk Pool....288
Set Disk Pool Complete....289
Set Enclosure Attribute....290
Set Enclosure Identification....290
Set Event Alert Filtering....291
Set Foreign Physical Disk to Native....292
Set Host....293
Set Host Channel....294
Set Host Group....295
Set Host Port....295
Set iSCSI Initiator....296
Set iSCSI Target Properties....297
Set Physical Disk Channel Status....297
Set Physical Disk Hot Spare....298
Set Physical Disk State....299
Set RAID Controller Module....299

Set Read-Only Snapshot Virtual Disk To A Read/Write Virtual Disk....303
Set Remote Replication....303
Set Remote Replication Group....306
Set Session....309
Set Snapshot Group Attributes....310
Set Snapshot Group Media Scan....311
Set Snapshot Group Repository Virtual Disk Capacity....312
Set Snapshot Group Schedule....313
Set Snapshot (Legacy) Virtual Disk....314
Set Snapshot Virtual Disk Media Scan....317
Set Snapshot Virtual Disk Repository Virtual Disk Capacity....318
Set Storage Array....319
Set Storage Array Enclosure Positions....322
Set Storage Array ICMP Response....323
Set Storage Array iSNS Server IPv4 Address....323
Set Storage Array iSNS Server IPv6 Address....324
Set Storage Array iSNS Server Listening Port....324
Set Storage Array Learn Cycle....325
Set Storage Array Redundancy Mode....325
Set Storage Array Security Key....326
Set Storage Array Time....326
Set Storage Array Unnamed Discovery Session....327
Set Thin Virtual Disk Attributes....327
Set Virtual Disk....329
Set Virtual Disk Attributes For A Disk Pool....332
Set Virtual Disk Copy....337
Set Virtual Disk Mapping....338
Show Blocked Events....340
Show Consistency Group Snapshot Image....340
Show Current iSCSI Sessions....341
Show Disk Group....342
Show Disk Group Export Dependencies....343
Show Disk Group Import Dependencies....343
Show Disk Pool....344
Show Host Ports....344
Show Physical Disk....345
Show Physical Disk Channel Statistics....347
Show Physical Disk Download Progress....347
Show RAID Controller Module....348
Show RAID Controller Module NVSRAM....349
Show Remote Replication Group....349
Show Remote Replication Group Synchronization Progress....350

Show Remote Replication Virtual Disk Candidates... 351
Show Remote Replication Virtual Disk Synchronization Progress... 352
Show SNMP Communities... 352
Show SNMP MIB II System Group Variables... 354
Show Snapshot Group... 354
Show Snapshot Image... 356
Show Snapshot Virtual Disks... 357
Show SSD Cache... 358
Show SSD Cache Statistics... 359
Show Storage Array... 363
Show Storage Array Auto Configure... 365
Show Storage Array Core Dump... 366
Show Storage Array DBM Database... 367
Show Storage Array Host Topology... 367
Show Storage Array LUN Mappings... 368
Show Storage Array Negotiation Defaults... 369
Show Storage Array ODX Setting... 369
Show Storage Array Power Information... 370
Show Storage Array Unconfigured iSCSI Initiators... 370
Show Storage Array Unreadable Sectors... 371
Show String... 371
Show Thin Virtual Disk... 372
Show Virtual Disk... 373
Show Virtual Disk Action Progress... 374
Show Virtual Disk Copy... 375
Show Virtual Disk Copy Source Candidates... 376
Show Virtual Disk Copy Target Candidates... 376
Show Virtual Disk Performance Statistics... 377
Show Virtual Disk Reservations... 377
Start Configuration Database Diagnostic... 378
Start Consistency Group Snapshot Rollback... 378
Start Disk Group Blink... 380
Start Disk Group Defragment... 380
Start Disk Group Export... 380
Start Disk Group Import... 381
Start Disk Pool Locate... 382
Start Enclosure Blink... 382
Start Input Output RAID Controller Module (IOC) Dump... 383
Start iSCSI DHCP Refresh... 383
Start Physical Disk Blink... 384
Start Physical Disk Channel Blink... 384
Start Physical Disk Initialize... 385

Start Physical Disk Reconstruction....385
Start Remote Replication Synchronization....386
Start Remote Replication (Legacy) Synchronization....386
Start Secure Physical Disk Erase....387
Start Snapshot Rollback....387
Start Snapshot Image Rollback....388
Start SSD Cache Performance Modeling....389
Start Storage Array Blink....390
Start Storage Array Configuration Database Diagnostic....390
Start Storage Array iSCSI Refresh....392
Start Virtual Disk Initialization....392
Stop Configuration Database Diagnostic....393
Stop Consistency Group Snapshot Rollback....393
Stop Consistency Group Snapshot Virtual Disk....394
Stop Disk Group Blink....395
Stop Disk Pool Locate....395
Stop Enclosure Blink....395
Stop Pending Snapshot Images On Consistency Group....396
Stop Physical Disk Blink....396
Stop Physical Disk Channel Blink....396
Stop Physical Disk Replace....397
Stop Snapshot Group Image....398
Stop Snapshot Group Pending Snapshot Images....398
Stop Snapshot (Legacy)....399
Stop Snapshot (Legacy) Rollback....400
Stop Snapshot Rollback....401
Stop Snapshot Image Rollback....401
Stop Snapshot Virtual Disk....402
Stop SSD Cache Performance Modeling....403
Stop Storage Array Blink....405
Stop Storage Array Configuration Database Diagnostic....405
Stop Storage Array iSCSI Session....406
Stop Storage Array Physical Disk Firmware Download....406
Stop Virtual Disk Copy....406
Suspend Remote Replication....407
Suspend Remote Replication Group....408
Suspend SSD Cache....408
Test Remote Replication Group Connectivity....409
Test SNMP Trap Destination....410
Unregister SNMP Community....411
Unregister SNMP Trap Destination....411
Update SNMP Community....411

Update SNMP MIB II System Group Variables....412
Update SNMP Trap Destination....413
Validate Storage Array Security Key....413

13 Sample Script Files....415
Configuration Script Example 1....415
Configuration Script Example 2....417

1

CLI Command Updates

This chapter reflects new and updated commands that are available for use with the Dell PowerVault MD 32XX/34XX/36XX/38XX Series storage arrays.

NOTE: Not all commands are valid with all storage arrays. Some commands are specific to certain platforms.

CAUTION: Script commands are capable of changing the configuration and may cause loss of data if not used correctly. Command operations are performed as soon as you run the commands. Before using the script commands, ensure that you have backed up all data, and have saved the current configuration so that you can reinstall it if the changes do not work.

New Commands

The following commands have been added to this guide to reflect the additional functionality available in the PowerVault MD32XX, MD34XX, MD36XX, and MD38XX.

- Recover SAS Port Mis-Wire
- Reduce Disk Pool Capacity
- Register SNMP Community
- Register SNMP Trap Destination
- Set Event Alert Filtering
- Show Blocked Events
- Show SNMP Communities
- Show SNMP MIB II System Group Variables
- Show Storage Array Power Information
- Test SNMP Trap Destination
- Unregister SNMP Community
- Unregister SNMP Trap Destination
- Update SNMP Community
- Update SNMP Trap Destination
- Update SNMP MIB II System Group Variables

Updated Commands

The syntax of the following commands has been modified, updated, or enhanced since the last release of this document.

NOTE: not all commands are valid with all storage arrays. Some commands are specific to certain platforms.

- Activate Remote Replication (Legacy)
- Download Storage Array NVSRAM
- Resume Remote Replication Group
- Resume Snapshot (Legacy) Rollback
- Set RAID Controller Module
- Set Remote Replication (Legacy)
- Set Storage Array
- Start Remote Replication Synchronization
- Stop Snapshot (Legacy) Rollback

2

About The Command Line Interface

This guide is intended for system administrators, developers, and engineers who need to use the command line interface (CLI) tool and its associated commands and script files. Selected CLI commands perform functions that can also be accessed from the Modular Disk (MD) Storage Manager, which is the graphical user interface (GUI) to the storage array. See the *Administrator's Guide*, which describes the Storage Manager software that is used to create and manage multiple storage arrays. For additional information, see the hardware and software manuals that shipped with your system.

NOTE: Always check for updates on **dell.com/support** and read the updates first because they often supersede information in other documents.

NOTE: CLI commands do not have interactive warnings for destructive commands.

CLI is a software tool that enables storage array installers, developers, and engineers to configure and monitor storage arrays. Using the command line interface, you can issue commands from an operating system prompt, such as the Microsoft Windows command prompt (`C:\`) or a Linux operating system terminal.

Each command performs a specific action for managing a storage array or returning information about the status of a storage array. You can enter individual commands, or run script files when you need to perform operations more than once (such as installing the same configuration on several storage arrays). A script file can be loaded and run from the command line interface. You can also run commands in an interactive mode. Interactive mode enables you to connect to a specific storage array and rapidly enter a command, determine the effect on the storage array, and then enter a new command.

The command line interface gives you direct access to a script engine utility in the Dell PowerVault Modular Disk Storage Manager software (MD Storage Manager). The script engine reads the commands, or runs a script file, from the command line and performs the operations instructed by the commands.

You can use the command line interface to perform the following functions:

- Directly access the script engine and run commands in interactive mode or using a script file.
- Create script command batch files to be run on multiple storage arrays when you need to install the same configuration on different storage arrays.
- Run script commands on a storage array directly connected to a host, a storage array connected to a host by an Ethernet, or a combination of both.
- Display configuration information about the storage arrays.
- Add storage arrays to and remove storage arrays from the management domain.
- Perform automatic discovery of all storage arrays attached to the local subnet.
- Add or delete Simple Network Management Protocol (SNMP) trap destinations and e-mail alert notifications.
- Specify the mail server and sender e-mail address or Simple Mail Transport Protocol (SMTP) server for alert notifications.
- Direct the output to a standard command line display or to a named file.

How To Use The Command Line Interface

Using the CLI commands, you can access the script engine, specify which storage array receives the script commands, and set operation environment parameters.

A CLI command consists of the following elements:

- The term **SMcli**
- Storage array identifier
- Parameters
- Script commands

The following syntax is the general form of a CLI command:

`SMcli` *`storageArray parameters script-commands;`*

where,

`SMcli` invokes the command line interface.

`storageArray` is the host name or IP address of the storage array.

`parameters` are the CLI parameters that define the environment and purpose of the command.

`script-commands` are the commands or name of the script file containing the script commands.

The script commands are the storage array configuration commands. About The Script Commands presents an overview of the script commands. Script commands provide definitions, syntax, and parameters for the script commands.

Usage Notes

If you enter **SMcli** and a storage array name but do not specify CLI parameters, script commands, or a script file, the command line interface runs in interactive mode. Interactive mode enables you to run individual commands without prefixing the commands with **SMcli**. You can enter a single command, view the results, and enter the next command without typing the complete **SMcli** string. Interactive mode is useful for determining configuration errors and quickly testing configuration changes.

If you enter **SMcli** without any parameters or with an incorrect parameter, the script engine returns usage information.

NOTE: The **SMcli** command is installed under the client directory of the selected path during a management station install of the MD Storage Manager software.

NOTE: The **SMcli** command should be a component of the system environment command path.

CLI Commands

This section lists the CLI commands you can use to perform the following functions:

- Identify storage arrays

- Set passwords
- Add storage arrays
- Specify communication parameters
- Enter individual script configuration commands
- Specify a file containing script configuration commands

The following are general forms of the CLI commands, showing the parameters and terminals used in each command. The table below lists definitions for the parameters shown in the CLI commands.

Table 1. Command Name Conventions

Parameter	Definition
`a|b`	pipe symbol indicating alternative ("a" or "b")
italicized-words	terminals
`[...]` (square brackets)	zero or one occurrence
`{...}` (curly brackets)	zero or more occurrences
`<...>` (angle brackets)	occurrence exceeds maximum limit of 30 characters
`(a|b|c)`	choose only one of the alternatives
bold	terminals

```
SMcli host-name-or-IP-address [host-name-or-IPaddress]
[-c "command; {command2};"]
[-n storage-array-name | -w WWID]
[-o outputfile][-p password][-e][-S]

SMcli host-name-or-IP-address
[host-name-or-IP-address] [-f scriptfile]
[-n storage-array-name | -w WWID]
[-o outputfile] [-p password] [-e] [-S]

SMcli (-n storage-array-name | -w WWID)
[-c "command; {command2};"]
[-o outputfile][-p password][-e][-S]

SMcli (-n storage-array-name | -w WWID)
[-f scriptfile]
[-o outputfile] [-p password] [-e] [-S]

SMcli (-n storage-array-name | -w WWID)
[-o outputfile][-p password][-e][-S]

SMcli -a email:email-address
[host-name-or-IP-address1
[host-name-or-IP-address2]]
[-n storage-array-name | -w WWID | -h host-name |
-r (host_sa | direct_sa)]
[-I information-to-include][-q frequency][-S]

SMcli -x email:email-address
[host-name-or-IP-address1
[host-name-or-IP-address2]]
[-n storage-array-name | -w WWID | -h host-name |
-r (host_sa | direct_sa)] [-S]

SMcli (-a | -x) trap:community,
host-name-or-IP-address [host-name-or-IP-address1
```

```
[host-name-or-IP-address2]]
[-n storage-array-name | -w WWID | -h host-name |
-r (host_sa | direct_sa)] [-S]

SMcli -d [-w][-i][-s][-v][-S]

SMcli -m host-name-or-IP-address -F email-address
[-g contactInfoFile][-S]

SMcli -A [host-name-or-IP-address
[host-name-or-IP-address]] [-S]

SMcli -X (-n storage-array-name | -w WWID |
-h host-name)

SMcli -?
```

Command Line Parameters

Table 2. Command Line Parameters

Parameter	Definition
host-name-or-IP-address	Specify either the host name or the Internet Protocol (IP) address of an in-band managed storage array (IPv4 or IPv6) or an out-of-band managed storage array (IPv4 or IPv6). • If you manage a storage array by using a host connected directly to the storage array (in-band storage management), you must use the -n parameter if more than one storage array is connected to the host. • If you manage a storage array through an Ethernet connection (out-of-band storage management), you must specify the host-name-or-IP-address of the redundant array of independent disks (RAID) controller modules. • If you have previously configured a storage array in the graphical user interface (GUI) of the MD Storage Manager, you can specify the storage array by its user-supplied name by using the -n parameter.
-A	Use to add a storage array to the configuration files. If you do not follow the -A parameter with a host-name-or-IP-address, automatic discovery scans the local subnet for storage arrays.
-a	Use to add an SNMP trap destination or an e-mail address alert destination. • When adding an SNMP trap destination, the SNMP community is automatically defined as the community name for the trap and the **host** is the IP address or Domain Name Server (DNS) host name of the system to which the trap should be sent. • When adding an e-mail address for an alert destination, the **email-address** is the e-mail address to which to send the alert message.

Parameter	Definition
`-c`	Use to indicate that you are entering one or more script commands to run on the specified storage array. Terminate each command by using a semicolon (;). You cannot place more than one `-c` parameter on the same command line. You can include more than one script command after the `-c` parameter.
`-d`	Use to display the contents of the script configuration file.
`-e`	Use to disable syntax checking when executing the current CLI command.
`-F (uppercase)`	Use to specify the e-mail address from which all alerts will be sent.
`-f (lowercase)`	Use to specify a file name containing script commands intended to run on the specified storage array. This parameter is similar to the `-c` parameter in that both are intended for running script commands. The `-c` parameter allows you to execute individual script commands. The `-f (lowercase)` parameter allows you to execute script commands contained in a file. **NOTE:** By default, any errors encountered when running the script commands in a file are ignored, and the file continues to run. To override this behavior, use the `set session errorAction=stop` command in the script file.
`-g`	Use to specify an ASCII file that contains e-mail sender contact information to include in all e-mail alert notifications. The CLI assumes the ASCII file is text only, without delimiters or any expected format. A typical file contains the following information: • Name • Title • Company • Phone • Pager **NOTE:** You can use any file name that your operating system supports. You must not use **userdata.txt**. Some operating systems reserve **userdata.txt** for system information.

Parameter	Definition
-h	Use with the -a and -x parameters to specify the host name that is running the SNMP agent to which the storage array is connected.
-I	Use to specify the type of information to be included in the e-mail alert notifications. The following are valid information arguments: • eventOnly— Only event information is included in the e-mail. • profile— Event and array profile information is included in the e-mail. • supportBundle— Event and support bundle information is included in the e-mail. NOTE: You can enter only one information argument each time you execute the command. If you want all of the information, you must run the command three times.
-i	Use with the -d parameter to display the IP address of the known storage arrays.
-m	Use to specify the host name or IP address of the e-mail server from which to send e-mail alert notifications.
-n	Use to specify the name of the storage array on which to run the script commands. This name is optional when you use host-name-or-IP-address; however, if you are using the in-band method for managing the storage array, you must use the -n parameter if more than one storage array is connected to the host at the specified address. The storage array name is required when host-name-or-IP-address is not used; however, the name of the storage array configured for use in the MD Storage Manager GUI (that is, listed in the configuration file) must not be a duplicate name of any other configured storage array.
-o	Use with the -c or -f parameter to specify a file name for all output text that is a result of running the script commands.
-p	Use to specify the password for the storage array on which to run commands. A password is not necessary under the following conditions: • A password has not been set on the storage array. • The password is specified in a script file that is running.

Parameter	Definition
	• The storage array password is specified by using the `-c` parameter and the `set session password=`*password* command.
`-q`	Use to specify how frequently to include additional profile or support bundle information in the e-mail alert notifications. An e-mail alert notification that contains at least the basic event information is always generated for every critical event. If you set the `-I` parameter to `eventOnly`, the only valid argument for `-q` is `everyEvent`. If you set the `-I` parameter to either `profile` or `supportBundle`, this information is included with the e-mails with the frequency specified by the `-q` parameter. Valid frequency arguments are: • `everyEvent` — Information is returned with every e-mail alert notification. • `2` — Information is returned no more than once every two hours. • `4` — Information is returned no more than once every four hours. • `8` — Information is returned no more than once every eight hours. • `12` — Information is returned no more than once every 12 hours. • `24` — Information is returned no more than once every 24 hours.
`-r`	Use with the `-a` or `-x` parameter to specify the name of a management station. The name of a management station can be either **direct_sa** (out-of-band storage array) or **host_sa** (in-band storage arrays [host-agent]). The `-r` parameter enables you to set or change the alert notifications for all storage arrays under each management station.
`-S (uppercase)`	Use to suppress the informational messages describing command progress that appear when running script commands. (Suppressing informational messages is also called *silent mode*.) This parameter suppresses the following messages: • Performance syntax check • Syntax check complete • Executing script • Script execution complete • SMcli completed successfully
`-s (lowercase)`	Use with the `-d` parameter to display the alert settings in the configuration file.

Parameter	Definition
`-v`	Use with the `-d` parameter to display the current global status of the known devices in the storage array configuration file. (The configuration file lists all of the devices in a storage array configuration and the relationship between the devices. Use the configuration file to reconstruct a storage array).
`-X (uppercase)`	Use to delete a storage array from the configuration file. (The configuration file lists all of the devices in a storage array configuration and the relationship between the devices. Use the configuration file to reconstruct a storage array).
`-x (lowercase)`	Use to remove an SNMP trap destination or an e-mail address alert destination. The **community** is the SNMP community name for the trap, and the **host** is the IP address or DNS host name of the system to which you want the trap sent.
`-?`	Use this parameter to display usage information about the CLI commands.

Formatting Considerations

Quotation marks (" ") used as part of a name or label require special consideration when you run the CLI and script commands on a Microsoft Windows operating system. The following explains the use of quotation marks in names while running CLI and script commands on Windows.

When quotation marks (" ") are part of an argument, you must insert a backslash (\) before each quotation mark character unless you are in interactive mode. For example:

```
-c "set storageArray userLabel=\"Engineering\";"
```

where, **Engineering** is the storage array name.

You cannot use quotation marks (" ") as part of a character string (also called string literal) within a script command. For example, you cannot enter the following string to set the storage array name to **"Finance"Array**:

```
-c "set storageArray userLabel=
\"\"Finance\"Array\";"
```

On a Linux operating system, the delimiters around names or labels are single quotation marks (' '). The Linux versions of the previous examples are:

```
-c 'set storageArray userLabel="Engineering";'
```

Detailed Error Reporting

Error data collected from an error encountered by the CLI is written to a file. Detailed error reporting under the CLI works as follows:

- If the CLI must abnormally end execution or abort script command execution, error data is collected and saved before the CLI aborts.

- The CLI automatically saves the error data by writing the data to a file with a standard name.
- The CLI does not have any provisions to avoid overwriting an existing version of the file containing error data.

For error processing, errors appear as two types:

- Parameter or syntax errors you might enter
- Exceptions that occur as a result of an operational error

When the CLI encounters either type of error, it writes information describing the error directly to the command line and sets a return code. Depending on the return code, the CLI might also write additional information about which parameter caused the error. The CLI also writes information about what command syntax was expected to help you identify any syntax errors you might have entered.

When an exception occurs while executing a command, the CLI automatically saves the error information to a file named **excprpt.txt**. The CLI attempts to place **excprpt.txt** in the directory specified by the system property **devmgr.datadir**, which by default is the **"client/data"** directory under the main installation directory in Windows and the **/var/opt/SM** directory in Linux. If for any reason the CLI cannot place the file in the **devmgr.datadir**-specified directory, the CLI saves the **excprpt.txt** file in the same directory from which the CLI is running. You cannot change the file name or location. The **excprpt.txt** file is overwritten every time an exception occurs. To save the information in the **excprpt.txt** file, you must copy the information to a new file or directory.

Exit Status

After you run a CLI command or a CLI and script command, status is displayed that indicates the success of the operation defined by the command. The status values are shown in the following table.

Table 3. Exit Status

Status Value	Meaning
0	The command terminated without an error.
1	The command terminated with an error. Error information is also displayed.
2	The script file does not exist.
3	An error occurred while opening an output file.
4	A storage array is not at the specified address.
5	Addresses specify different storage arrays.
6	A storage array name does not exist for the host agent connected.
7	The storage array name was not at the specified address.
8	The storage array name was not in the configuration file.
10	A management class does not exist for the storage array.
11	A storage array was not found in the configuration file.
12	An internal error occurred.
13	Invalid script syntax was found.

Status Value	Meaning
14	The RAID controller module was unable to communicate with the storage array.
15	A duplicate argument was entered.
16	An execution error occurred.
17	A host was not at the specified address.
18	The World Wide Identifier (WWID) was not in the configuration file.
19	The WWID was not at the address.
20	An unknown IP address was specified.
21	The event monitor configuration file was corrupted.
22	The storage array was unable to communicate with the event monitor.
23	The RAID controller module was unable to write alert settings.
24	The wrong management station was specified.
25	The command was not available.
26	The device was not in the configuration file.
27	An error occurred while updating the configuration file.
28	An unknown host error occurred.
29	The sender contact information file was not found.
30	The sender contact information file could not be read.
31	The **userdata.txt** file exists.
32	An invalid *-l* value in the e-mail alert notification was specified.
33	An invalid *-f* value in the e-mail alert notification was specified.

Usage Examples

The following examples show how to enter CLI commands on a command line. The examples show the syntax, form, and in some examples, script commands. Examples are shown for both Windows and Linux operating systems. The usage for the *-c* parameter varies depending on your operating system. On Windows operating systems, put quotation marks (" ") around the script command following the *-c* parameter. On Linux operating systems, put single quotation marks (' ') around the script command following the *-c* parameter.

NOTE: See Script Commands for descriptions of the script commands used in the following examples.

This example shows how to change the name of a storage array. The original name of the storage array is **Payroll_Array**. The new name is **Finance_Array**.

Windows:

```
SMcli -n "Payroll_Array" -c "set storageArray userLabel=\"Finance_Array\";"
```

Linux:

```
SMcli -n 'Payroll_Array' -c 'set storageArray userLabel="Finance_Array";'
```

This example shows how to delete an existing virtual disk and create a new virtual disk on a storage array. The existing virtual disk name is **Stocks_<_Bonds**. The new virtual disk name is Finance. The RAID controller module host names are **finance1** and **finance2**. The storage array is protected and requires the password **TestArray**.

Windows:

```
SMcli finance1 finance2 -c "set session password= \"TestArray\"; delete
virtualDisk [\"Stocks_<_Bonds\"];create virtualDisk physicalDiskCount[3]
raidLevel=5 capacity=10 GB userLabel=\"Finance\"; show storageArray
healthStatus;"
```

Linux:

```
SMcli finance1 finance2 -c 'set session password= delete virtualDisk
["Stocks_<_Bonds"]; create virtualDisk physicalDiskCount[3] raidLevel=5
capacity=10 GB userLabel="Finance"; show storageArray healthStatus;'
```

This example shows how to run commands in a script file named **scriptfile.scr** on a storage array named **Example**. The *-e* parameter runs the file without checking syntax. Executing an `SMcli` command without checking syntax enables the file to run more quickly; however, the `SMcli` command may not execute correctly if the syntax is incorrect.

```
SMcli -n Example -f scriptfile.scr -e
```

This example shows how to run commands in a script file named **scriptfile.scr** on a storage array named **Example**. In this example, the storage array is protected by the password **My_Array**. Output, as a result of commands in the script file, goes to file **output.txt**.

Windows:

```
SMcli -n Example -f scriptfile.scr -p "My_Array" - o output.txt
```

Linux:

```
SMcli -n Example -f scriptfile.scr -p 'My_Array' - o output.txt
```

This example shows how to display all storage arrays that are currently discovered in the current configuration. The command in this example returns the host name of each storage array.

```
SMcli -d
```

If you want to know the IP address of each storage array in the configuration, add the *-i* parameter to the command.

```
SMcli -d -i
```

3

About The Script Commands

You can use the script commands to configure and manage a storage array. The script commands are distinct from the command line interface (CLI) commands; however, you enter the script commands using the command line interface. You can enter individual script commands, or run a file of script commands. When entering an individual script command, include it as part of a CLI command. When running a file of script commands, include the file name as part of a CLI command. The script commands are processed by a script engine that performs the following functions:

- Verifies command syntax
- Interprets the commands
- Converts the commands to the appropriate protocol-compliant commands, which is, in turn, run by the RAID controller module
- Passes the commands to the storage array

At the storage array, the redundant array of independent disks (RAID) controller modules in the storage array runs the script commands.

The script engine and script commands support the storage array configuration and management operations listed in the following table.

Table 4. Configuration and Management Operations

Operation	Activities
Virtual disk, disk group configuration	Creating, deleting, and setting priority; labeling; setting physical disk composition when creating virtual disks; setting segment size; and setting media scan control
Physical disk configuration	Configuring the hot spare
RAID controller module configuration	Defining virtual disk ownership, changing mode settings, defining network settings, and setting host port IDs
General storage array configuration	Resetting a configuration to defaults, labeling, checking the health status, setting the time of day, clearing the Major Event Log, and setting the media scan rate
NVSRAM configuration	Downloading and modifying the user configuration region at the bit and byte level, displaying nonvolatile static random access memory (NVSRAM) values
Product identification	Retrieving the enclosure profile display data
Battery management	Setting the battery installation date

Operation	Activities
Firmware management	Downloading RAID controller module, enclosure management module (EMM), and physical disk firmware

Script Command Structure

All script commands have the following structure:

```
command operand-data {statement-data}
```

where, *command* identifies the action to be performed, *operand-data* represents the storage array component to configure or manage (such as a RAID controller module, physical disk, or disk group), and *statement-data* is what you want to do to the component (such as, specifying the RAID level or availability of a disk group).

The general form of the syntax for operand-data is as follows:

```
(object-type | allobject-types | [qualifier]
(object-type [identifier] {object-type
[identifier]} | object-types [identifier-list]))
```

An *operand-data* object can be identified four ways:

- The object types and object qualifiers
- The `all` parameter
- Brackets
- A list of identifiers

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Use an object type when the command is not referencing a specific object. The `all` parameter means all objects of the specified type in the storage array (for example, *allVirtualDisks*).

To perform a command on a specific object, use brackets to identify the object (for example, **virtualDisk[engineering]**). Specify a subset of objects with a list of identifiers in brackets (for example, **virtualDisks[sales engineering marketing]**). In a list of identifiers, use a blank space as the delimiter. A qualifier is necessary if you want to include additional information to describe the objects.

The following table lists the object type and identifiers associated with the object types.

Table 5. Object Types and Identifiers

Object Type	Identifier
controller	0 or 1
physicalDisk	Enclosure ID and the slot ID
physicalDiskChannel	Physical disk channel identifier
GroupName	Remote Replication virtual disk user label

Object Type	Identifier
host	User label
hostChannel	Host channel identifier
hostGroup	User label
hostPort	User label
snapVirtualDiskName	Virtual disk user label
snapshot	Virtual disk user label
snapGroup	A snapshot group contains a sequence of snapshot images of an associated base virtual disk. A snapshot group has a repository virtual disk that is used to save data for all of the snapshot images that are part of the snapshot group.
snapGroupName	Virtual disk group user label
storageArray	Not applicable
enclosure	Enclosure ID
virtualDisk	Virtual disk user label or the World Wide Identifier (WWID) for the virtual disk (set command only)
virtualDiskCopy	Target virtual disk and, optionally, the source virtual disk user labels
diskGroup	Virtual disk group number

Statement data is in the form of **attribute=value** (such as **raidLevel=5**), an **attribute name** (such as **batteryInstallDate**), or an **operation name** (such as **consistencyCheck**).

Script Command Synopsis

Because you can use the script commands to define and manage the different aspects of a storage array (such as host topology, physical disk configuration, RAID controller module configuration, virtual disk definitions, and disk group definitions), the actual number of commands is extensive. The commands, however, fall into general categories that are reused when you apply the commands to the different aspects of a storage array.

The following table lists the general form of the script commands and provides a definition of each command.

Table 6. General Form of the Script Commands

Command Syntax	Description
activate *object* {*statement-data*}	Sets up the environment so that an operation can take place or performs the operation if the environment is already correctly set up.
autoConfigure *storageArray*{*statement-data*}	Automatically creates a configuration based on parameters specified in the command.

Command Syntax	Description
check *object* {*statement*-data}	Starts a synchronous operation to report on errors in the object.
clear *object* {*statement-data*}	Discards the contents of some attribute of an object. This is a destructive operation that cannot be reversed.
create *object* {statement-*data*}	Creates an object of the specified type.
deactivate *object* {*statement-data*}	Removes the environment for an operation.
delete *object*	Deletes a previously created object.
diagnose *object* {*statement-data*}	Runs a test and displays the results.
disable *object* {*statement-data*}	Prevents a feature from operating.
download *object* {*statement-data*}	Transfers data to the storage array or hardware associated with the storage array.
enable *object* {*statement-data*}	Allows a feature to operate.
recopy *object* {*statement-data*}	Restarts a virtual disk copy operation by using an existing virtual disk copy pair. You can change attributes before the operation is restarted.
recover *object* {*statement-data*}	Re-creates an object from saved configuration data and the statement attributes (similar to the **create** command).
recreate *object* {*statement-data*}	Restarts a snapshot operation using an existing snapshot virtual disk. You can change attributes before the operation is restarted.
remove *object* {*statement-data*}	Removes a relationship from between objects.
repair *object* {*statement-data*}	Repairs errors found by the check command.
reset *object* {*statement-data*}	Returns the hardware or object to an initial state.
resume *object*	Starts a suspended operation. The operation begins where it left off when suspended.
revive *object*	Forces the object from the Failed to the Optimal state. Use only as part of an error recovery procedure.
save *object* {*statement-data*}	Writes information about the object to a file.
set *object* {*statement-data*}	Changes object attributes. All changes are completed when the command returns.
show *object* {*statement-data*}	Displays information about the object.
start *object* {*statement-data*}	Starts an asynchronous operation. You can stop some operations after they have started. You can query the progress of some operations.

Command Syntax	Description
stop *object* {*statement-data*}	Stops an asynchronous operation.
suspend *object* {*statement-data*}	Suspends an operation. You can then restart the suspended operation, and it continues from the point at which it was suspended.

Recurring Syntax Elements

Recurring syntax elements are a general category of variables and parameters you can use in one or more script commands. The recurring syntax is used in the general definitions of the script commands that are listed in Script Commands. The following table lists the recurring syntax and the syntax values that you can use with the syntax.

Table 7. Recurring Syntax Elements

Recurring Syntax	Syntax Value
raid-level	(0 | 1 | 5| 6)
snapshot-repository-raid-level	(1 | 5 | 6)
capacity-spec	*integer-literal* [KB | MB | GB | TB | Bytes]
segment-size-spec	integer-literal
boolean	(TRUE | FALSE)
user-label	string-literal
user-label-list	*user-label* {*user-label*}
create-raid-vol-attr-value-list	*create-raid-virtual disk-attribute-valuepair* {*create-raid-virtual disk-attribute-valuepair*}
create-raid-virtual disk-attribute-value-pair	capacity=*capacity-spec* | owner=(0 | 1) | segmentSize=*integer-literal*
RAID controller module-enclosureId	(0–99)
slot-id	(0–31)
port-id	(0–127)
physical disk-spec	enclosureID, slotID
physical disk-spec-list	*physical disk-spec* {*physical disk-spec*}
enclosure-id-list	*enclosureID* {*enclosureID*}
hex-literal	0x hexadecimal-literal
virtual disk-group-number	integer-literal
filename	string-literal
error-action	(stop | continue)
physical disk-channel-identifier	(1 | 2)

Recurring Syntax	Syntax Value
physical disk-channel-identifier-list	*physical disk-channel-identifier* {*physical disk-channel-identifier*}
host-channel-identifier	(01 | 02 | 11 | 12)
physical disk-type	(Serial Attached SCSI [SAS])
feature-identifier	(snapshot | virtualDiskCopy)
repository-spec	instance-based-repository-spec | count-based-repository-spec
ethernet-port-options	**IPV4Address**=*ipv4-address*| **IPV4ConfigurationMethod**=[*(static* | *dhcp)*] | **IPV4GatewayIP**=*ipv4-address*| **IPV4SubnetMask**=*ipv4-address*|
iscsi-host-port-options	**IPV4Address**=*ipv4-address* | **IPV6LocalAddress**=*ipv6-address* | **IPV6RoutableAddress**=*ipv6-address* | **IPV6RouterAddress**=*ipv6-address* | **enableIPV4**= *boolean* | **enableIPV6**=*boolean* | **enableIPV4Vlan**=*boolean* | **enableIPV6Vlan**=*boolean* | **enableIPV6Priority**=*boolean* | **enableIPV6Priority**=*boolean* | **IPV4ConfigurationMethod**=(*static* | *dhcp*) | **IPV6ConfigurationMethod**=(*static* | *auto*) | **IPV4GatewayIP**= *ipv4-address* | **IPV6HopLimit**=*integer* | **IPV6NdDetectDuplicateAddress**= *integer* | **IPV6NdReachableTime**=*integer* | **IPV6NdRetransmitTime**=*integer* | **IPV6NdTimeOut**=*integer* | **IPV4Priority**=*integer* | **IPV6Priority**=*integer* |

Recurring Syntax	Syntax Value
	IPV4SubnetMask=*ipv4-address* |
	IPV4VlanID=*integer* |
	IPV6VlanID=*integer* |
	maxFramePayload=*integer* |
	tcpListeningPort=*tcp-port-id*|
	NOTE: You must set the enableIPV4 parameter or the enableIPV6 parameter to TRUE to ensure that the specific IPV4 or IPV6 setting is applied.
	NOTE: The IPV6 address space is 128 bits. It is represented by eight 16-bit hexadecimal blocks separated by colons. You may drop leading zeros, and use a double colon to represent consecutive blocks of zeroes.
instance-based-repository-spec	repositoryRAIDLevel=*repository-raidlevel*
	repositoryPhysicalDisks=(*physical diskspec- list*)
	[enclosureLossProtect=boolean] repositoryDiskGroup=virtual-diskgroup- number [freeCapacityArea=integer-literal]
	Specify `repositoryRAIDLevel` with `repositoryPhysicalDisks`. Do not specify RAID level or physical disks with a disk group. Do not set `enclosureLossProtect` when specifying a disk group.
	NOTE: For enclosure loss protection to work, each physical disk in a disk group must be on a separate enclosure. If you set `enclosureLossProtect=TRUE` and have selected more than one physical disk from any one enclosure, the storage array returns an error. If you set `enclosureLossProtect=FALSE`, the storage array performs operations, but the disk group you create might not have enclosure loss protection.
	NOTE: To determine if a free capacity area exists, issue the `show diskGroup` command.
count-based-repository-spec	repositoryRAIDLevel=*repository-raidlevel*
	repositoryPhysicalDiskCount=*integerliteral*

Recurring Syntax	Syntax Value
	[physicalDiskType=physical disk-type]
	[enclosureLossProtect=boolean]
WWID	string-literal
	For hostPort identifiers this is a 16-digit hex number without any colon delimiters.
nvsram-offset	hexadecimal-literal
host-type	string-literal | integer-literal
nvsram-byte-setting	nvsram-value (0x hexadecimal | integer-literal)
nvsram-bit-setting	nvsram-mask, nvsram-value (0x hexadecimal, 0x hexadecimal | integer-literal)
ipv4-address	(0–255).(0–255).(0–255).(0–255)
ipv6-address	(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-FFFF):
	(0-FFFF):(0-FFF)
autoconfigure-vols-attr-value-list	*autoconfigure-vols-attr-value-pair* {*autoconfigure-vols-attr-value-pair*}
autoconfigure-vols-attr-value-pair	physicalDiskType=physical disk-type| raidLevel=raid-level |
	diskGroupWidth=integer-literal |
	diskGroupCount=integer-literal |
	virtualDisksPerGroupCount=*integerliteral*|
	hotSpareCount=*integer-literal* |
	segmentSize=*segment-size-spec*
	NOTE: The `physicalDiskType` parameter is not required if only one type of physical disk is in the storage array. If you use the `physicalDiskType` parameter, you must also use the `hotSpareCount` and `diskGroupWidth` parameters. If you do not use the `physicalDiskType` parameter, the configuration defaults to SAS physical disks.

Recurring Syntax	Syntax Value
	NOTE: The `virtualDisksPerGroupCount` parameter is the number of equal capacity virtual disks per disk group.
create-virtual-disk-copy-attr-value-list	create-virtual-disk-copy-attr-value-pair {*create-virtual-disk-copy-attr-value-pair*}
create-virtual-disk-copy-attr-value-pair	copyPriority=highest | high | medium | low | lowest | targetReadOnlyEnabled=boolean
recover-raid-virtual-disk-attr-value-list	recover-raid-virtual-disk-attr-value-pair {*recover-raid-virtual-disk-attr-value-pair*}
recover-raid-virtual-disk-attr-value-pair	owner=(0 | 1)

Table 8. Range of Values for Recurring Syntax Elements

Recurring Syntax	Syntax Values
IPV4Priority	0 to 7
IPV4VlanID	1 to 4094
IPV6Priority	0 to 7
IPV6VlanID	1 to 4094
IPV6HopLimit	0 to 255 (default value is 64)
IPV6NdDetectDuplicateAddress	0 to 256
IPV6NdReachableTime	0 to 65535 (default value is 30000 milliseconds)
IPV6RetransmitTime	0 to 65535 (default value is 1000 milliseconds)
IPV6NDTimeOut	0 to 65535 (default value is 3000 milliseconds)
maxFramePayload	1500 NOTE: The `maxFramePayload` parameter is shared between IPv4 and IPv6. The payload portion of a standard Ethernet frame is set at 1500 bytes, and a jumbo Ethernet frame is set at 9000 bytes. When using jumbo frames, make sure that all of the devices contained in the network path can handle the larger frame size.
tcpListeningPort (tcp-port-id)	3260, or 49,152 to 65,536 The default value is 3260.

Usage Guidelines

The following list provides guidelines for writing script commands on the command line:

- You must end all commands with a semicolon (;).
- You can enter more than one command on a line, but you must separate each command with a semicolon (;).
- You must separate each base command and its associated primary and secondary parameters with a space.
- The script engine is case sensitive.
- You can add comments to your scripts to make it easier for you and future users to understand the purpose of the script commands. For information on how to add comments, see Adding Comments To A Script File.

NOTE: User labels (such as for virtual disk, hosts, or host ports) are case sensitive. If you map to an object identified by a user label, you must enter the user label exactly as it is defined, or the CLI and script commands fails.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

NOTE: The `capacity` parameter returns an error if you specify a value greater than or equal to 10 without a space separating the numeric value and its unit of measure. For example, `10 GB` returns an error, but `9 GB` does not return an error.

Adding Comments To A Script File

You can add comments to a script file in three ways:

- The script engine interprets as a comment any text typed after two forward slashes (/ /) until an end-of-line character is reached. If the script engine does not find an end-of-line character in the script after processing a comment, an error message is displayed, and the script operation is terminated. This error commonly occurs when a comment is placed at the end of a script and you have not pressed **<Enter>**.

  ```
  // Deletes the existing configuration.
  clear storageArray Configuration;
  ```

- The script engine interprets any text typed between / * and * / as a comment. If the script engine does not find both a beginning and ending comment notation, an error message is displayed, and the script operation is terminated.

  ```
  /* Deletes the existing configuration */
  clear storageArray Configuration;
  ```

- Use the **show** statement to embed comments in a script file that you want to display while the script file is running. Enclose the text you want to display in quotation marks (" ").

  ```
  show "Deletes the existing configuration";
  clear storageArray Configuration;
  ```

4

Configuring A Storage Array

This chapter explains how to run script commands from the command line to create a virtual disk from a group of physical disks, and how to configure a redundant array of independent disks (RAID) storage array. This chapter assumes that you understand basic RAID concepts and terminology. Before configuring the storage array, become familiar with the concepts of physical disks, disk groups, virtual disks, host groups, hosts, and RAID controller modules. Additional information about configuring a storage array and related definitions is in the online help, the *Deployment Guide*, the *MD Storage Manager online help*, and the *Owner's Manual*.

Configuring a RAID storage array requires caution and planning to ensure that you define the correct RAID level and configuration for your storage array. The main purpose in configuring a storage array is to create virtual disks addressable by the hosts from a collection of physical disks. The commands described in this chapter enable you to set up and run a RAID storage array. Additional commands are also available to provide more control and flexibility. Many of these commands, however, require a deeper understanding of the firmware as well as various structures that need to be mapped. Use all the command line interface (CLI) commands and script commands with caution.

The following sections in this chapter show some, but not all, of the CLI and script commands. The purpose of showing these commands is to explain how you can use the commands to configure a storage array. The presentation in this chapter does not explain all possible usage and syntax for the commands. For complete definitions of the commands, including syntax, parameters, and usage notes, see Script Commands.

This chapter contains examples of CLI and script command usage. The command syntax used in the examples is for a host running a Microsoft Windows operating system. As part of the examples, the complete `C:\` prompt and DOS path for the commands are shown. Depending on your operating system, the prompt and path construct varies.

For most commands, the syntax is the same for all Windows and Linux operating systems, as well as for a script file. Windows operating systems, however, have an additional requirement when entering names in a command. On Windows, you must enclose the name between two backslashes (\) in addition to other delimiters. For example, the following name is used in a command that runs under Windows:

```
[\"Engineering\"]
```

For a Linux system when used in a script file, the name appears as:

```
["Engineering"]
```

Configuring A Storage Array

When you configure a storage array, you can maximize data availability by ensuring that data is quickly accessible while maintaining the highest level of data protection possible. The speed at which a host can access data is affected by the disk group RAID level and the segment size settings. Data protection is determined by the RAID level, hardware redundancy (such as global hot spares), and software redundancy (such as the Snapshot feature).

In general, you configure a storage array by defining the following entities:

- A disk group and associated RAID level
- The virtual disks
- Which hosts have access to the virtual disks

This section explains how to use the script commands to create a configuration from an array of physical disks.

Determining What Is On Your Storage Array

Even when you create a configuration on a previously unconfigured storage array, you still need to determine the hardware and software features that must be included with the storage array. When you configure a storage array with an existing configuration, you must ensure that your new configuration does not inadvertently alter the existing configuration, unless you are reconfiguring the entire storage array. For example, to create a new disk group on unassigned physical disks, you must determine which physical disks are available. The commands described in this section enable you to determine the components and features in your storage array.

The `show storageArray` command returns the following general information about the components and properties of the storage array:

- A detailed profile of the components and features in the storage array
- The battery age
- The default host type (which is the current host type)
- Other available host types
- The hot spare locations
- The identifiers for enabled features
- The logical and physical component profiles
- The time to which both RAID controller modules are set
- The RAID controller module that currently owns each virtual disk in the storage array

To return the most information about the storage array, run the `show storageArray` command with the `profile` parameter. The following is an example of the complete CLI and script command running on Windows:

```
client>smcli 123.45.67.89 -c "show storageArray profile;"
```

This example identifies the storage array by the dummy IP address 123.45.67.89. You can also identify the storage array by name.

The `show storageArray profile` command returns detailed information about the storage array. The information is presented in several screens on a display. You might need to increase the size of your

display buffer to see all of the information. Because this information is so detailed, you might want to save the output to a file. To save the output to a file, enter the command as shown in the following example:

```
client>smcli 123.45.67.89 -c "show storageArray profile;"
 -o c:\folder\storageArrayprofile.txt
```

In this example, the name *folder* is the folder in which you choose to place the profile file, and **storageArrayprofile.txt** is the name of the file. You can choose any folder and any file name.

CAUTION: When you write information to a file, the script engine does not check to determine if the file name already exists. If you choose the name of a file that already exists, the script engine writes over the information in the file without warning.

When you save the information to a file, you can use the information as a record of your configuration and as an aid during recovery.

To return a brief list of the storage array features and components, use the `summary` parameter. The command is similar to the following example:

```
client>smcli 123.45.67.89 -c "show storageArray summary;"
```

The summary information is also returned as the first section of information when you use the `profile` parameter.

The following `show` commands return information about the specific components of a storage array. The information returned by each of these commands is the same as the information returned by the `show storageArray profile` command, but is constrained to the specific component.

NOTE: The following commands are not complete commands. For information about a command, see the referenced section next to the command.

- `show controller` (RAID Controller Module Commands)
- `show physicalDisks` (Show Physical Disk)
- `show physicalDiskchannels stats` (Show Physical Disk Channel Statistics)
- `show storageArray hostTopology` (Show Storage Array Host Topology)
- `show storageArray lunmappings` (Show Storage Array LUN Mappings)
- `show allVirtualDisks` (Show Virtual Disk)
- `show diskGroup` (Show Disk Group)
- `show virtualDisk reservations` (Show Virtual Disk Reservations)

The following commands also return information about a storage array:

- `show storageArray autoConfigure` (Show Storage Array Auto Configure)
- `show controller NVSRAM` (Show RAID Controller Module NVSRAM)
- `show storageArray unreadableSectors` (Show Storage Array Unreadable Sectors)
- `show virtualDiskCopy sourceCandidates` (Show Virtual Disk Copy Source Candidates)
- `show virtualDiskCopy targetCandidates` (Show Virtual Disk Copy Target Candidates)
- `show virtualDisk performanceStat` (Show Disk Group Import Dependencies)

For descriptions of the show commands, including examples of the information returned by each command, see Script Commands. Other commands can help you learn about your storage array. To see a list of the commands, see Commands Listed By Function. These commands are organized by the

storage array activities that the commands support. Examples include virtual disk commands, host commands, enclosure commands, and others.

Saving A Configuration To A File

CAUTION: When you write information to a file, the script engine does not check to determine if the file name already exists. If you choose the name of a file that already exists, the script engine writes over the information in the file without warning.

After you have created a new configuration or if you want to copy an existing configuration for use on other storage arrays, you can save the configuration to a file. To save the configuration, use the `save storageArray configuration` command. Saving the configuration creates a script file that you can run on the command line. The following syntax is the general form of the command:

```
save storageArray configuration file="filename"
[(allconfig | globalSettings=(TRUE | FALSE)) |
virtualDiskConfigAndSettings=(TRUE | FALSE) |
hostTopology=(TRUE | FALSE) | lunMappings=(TRUE |
FALSE)]
```

You can choose to save the entire configuration or specific configuration features. The command for setting this parameter value looks like the following example:

```
client>smcli 123.45.67.89 -c "save storageArray configuration file=
\"c:\folder\storageArrayconfig1.scr\";"
```

In this example, the name **folder** is the folder in which you choose to place the configuration file, and **storageArrayconfig1.scr** is the name of the file. Choose any folder and any file name. MD Storage Manager uses the file extension **.scr** when it creates the configuration file.

Using The Create Virtual Disk Command

The `create virtualDisk` command enables you to create new virtual disks in the storage array in three ways:

- Create a new virtual disk while simultaneously creating a new disk group to which you assign the physical disks.
- Create a new virtual disk while simultaneously creating a new disk group to which the MD Storage Manager software assigns the physical disks.
- Create a new virtual disk in an existing disk group.

You must have unassigned physical disks in the disk group. You do not need to assign the entire capacity of the disk group to a virtual disk.

Creating Virtual Disks With User-Assigned Physical Disks

When you create a new virtual disk and assign the physical disks to use, the MD Storage Manager software creates a new disk group. The RAID controller module firmware assigns a disk group number to the new disk group. The following syntax is the general form of the command:

```
create virtualDisk physicalDisks=
(enclosureID0,slotID0...enclosureIDn,slotIDn)
raidLevel=(0 | 1 | 5 | 6)userLabel=
"virtualDiskName" [capacity=virtualDiskCapacity
owner=(0 | 1) segmentSize=segmentSizeValue]
[enclosureLossProtect=(TRUE | FALSE)]
```

NOTE: The `capacity`, `owner`, `segmentSize`, and `enclosureLossProtect` parameters are optional. You can use one or all of the optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

The `userLabel` parameter is the name to give to the virtual disk. The virtual disk name can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the virtual disk name is 30 characters. Spaces are not allowed. You must put quotation marks (" ") around the virtual disk name.

The `physicalDisks` parameter is a list of the physical disks that you want to use for the disk group. Enter the enclosure ID and slot ID of each physical disk that you want to use. Put parentheses around the list. Separate the enclosure ID and slot ID of a physical disk by a comma. Separate each enclosure ID and slot ID pair by a space. For example:

```
(0,0 0,1 0,2 0,3 0,4)
```

The `capacity` parameter defines the size of the virtual disk. You do not have to assign the entire capacity of the physical disks to the virtual disk. You can later assign any unused space to another virtual disk.

The `owner` parameter defines the RAID controller module to which you want to assign the virtual disk. If you do not specify a RAID controller module, the RAID controller module firmware determines the owner of the virtual disk.

The `segmentSize` parameter is the same as described for the `autoConfigure storageArray` command. See Using The Auto Configure Command.

The `enclosureLossProtect` parameter turns on or turns off enclosure loss protection for the disk group. For a description of how enclosure loss protection works, see Enclosure Loss Protection.

Example Of Creating Virtual Disks With User-Assigned Physical Disks

```
client>smcli 123.45.67.89 -c "create virtualDisk
physicalDisks=(0,0 0,1 0,2) raidLevel=5 userLabel=
\"Engineering_1\" capacity=20 GB owner=0;"
```

NOTE: The `capacity` parameter returns an error if you specify a value greater than or equal to 10 without a space separating the numeric value and its unit of measure. (For example, `10 GB` returns an error, but `10 GB` does not return an error).

The command in this example automatically creates a new disk group and a virtual disk with the name **Engineering_1**. The disk group has a RAID level of 5 (RAID 5). The command uses three physical disks to construct the disk group. The virtual disk created has a capacity of 20 GB. If each physical disk has a capacity of 73 GB, the total capacity of the disk group is 219 GB.

Because only 20 GB are assigned to the virtual disk, 199 GB remain available for other virtual disks that you can later add to this disk group. The segment size for each virtual disk is 64 KB. Hot spares have not been created for this new disk group. You must create hot spares after running this command.

Creating Virtual Disks With Software-Assigned Physical Disks

You can let the MD Storage Manager software assign the physical disks when you create the virtual disk. To have the software assign the physical disks, only specify the number of physical disks to use. The MD Storage Manager software then chooses the physical disks on which the virtual disk is created. The RAID

controller module firmware assigns a disk group number to the new disk group. The following syntax is the general form for the command:

```
create virtualDisk physicalDiskCount=
numberOfPhysicalDisks raidLevel=(0 | 1 | 5 | 6)
userLabel="virtualDiskName" [physicalDiskType=
(SAS)] [capacity=virtualDiskCapacity | owner=(0 |
1) | segmentSize=segmentSizeValue]
[enclosureLossProtect=(TRUE | FALSE)])
```

NOTE: The `physicalDiskType`, `capacity`, `owner`, `segmentSize`, and `enclosureLossProtect` parameters are optional. You can use one or all of the optional parameters as needed to help define your configuration. You do not, however, need to use any optional parameters.

This command is similar to the previous `create virtualDisk` command, which allows the user to assign the physical disks. This version of the command requires only the number and the type of physical disks to use in the disk group. You do not need to enter a list of physical disks. All other parameters are the same. Enclosure loss protection is performed differently when MD Storage Manager assigns the physical disks as opposed to when a user assigns the physical disks. For an explanation of the difference, see Enclosure Loss Protection.

Example Of Creating Virtual Disks With Software-Assigned Physical Disks

```
client>smcli 123.45.67.89 -c "create virtualDisk
physicalDiskCount=3 raidLevel=5 userLabel=
\"Engineering_1"\ capacity=20 GB owner=0
segmentSize=64;"
```

The command in this example creates the same virtual disk as the previous `create virtualDisk` command. However, in this case the user does not know which physical disks are assigned to this disk group.

Creating Virtual Disks In An Existing Disk Group

To add a new virtual disk to an existing disk group, use the following command:

```
create virtualDisk DiskGroup=diskGroupNumber
userLabel="virtualDiskName" [freeCapacityArea=
freeCapacityIndexNumber | capacity=
virtualDiskCapacity | owner=(0 | 1) | segmentSize=
segmentSizeValue]
```

NOTE: The `freeCapacityArea`, `capacity`, `owner`, and `segmentSize` parameters are optional. You can use one or all optional parameters as needed to help define your configuration, though you do not need to use any of them.

The `diskGroup` parameter is the number of the disk group in which you want to create a new virtual disk. If you do not know the disk group numbers on the storage array, you can use the `show allVirtualDisks summary` command. This command displays a list of the virtual disks and the disk groups to which the virtual disks belong.

The `userLabel` parameter is the name you want to give to the virtual disk. The virtual disk name can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the virtual disk name is 30 characters. You must enclose the virtual disk name with quotation marks (" ").

The `freeCapacityArea` parameter defines the free capacity area to use for the virtual disk. If a disk group has several free capacity areas, you can use this parameter to identify which free capacity area to

use for virtual disk creation. You do not have to assign the entire capacity of the physical disks to the virtual disk. Assign any unused space to another virtual disk at another time.

The `userLabel`, `capacity`, `owner`, and `segmentSize` parameters are the same as in the previous versions of the `create virtualDisk` command.

Enclosure Loss Protection

The `enclosureLossProtect` parameter is a boolean switch that turns enclosure loss protection on or off. To work properly, each physical disk in a virtual disk group must be in a separate enclosure. Enclosure loss protection is set under the following conditions:

- You assign the physical disks.
- The RAID controller module assigns the physical disks.

The following table shows possible results for the `enclosureLossProtect` parameter. The results depend on whether you assign the physical disks or the RAID controller module assigns the physical disks.

Method	enclosureLossProtect= TRUE	enclosureLossProtect=FALSE
You assign the physical disks.	If you select more than one physical disk from any one enclosure, the storage array returns an error.	The storage array performs the operation, but the created disk group does not have enclosure loss protection.
The RAID controller module firmware assigns the physical disks.	The storage array posts an error if the RAID controller module firmware cannot provide physical disks to ensure that the new disk group has enclosure loss protection.	The storage array performs the operation even if it means that the disk group might not have enclosure loss protection.

The `enclosureLossProtect` parameter is not valid when creating virtual disks on existing disk groups.

Using The Auto Configure Command

The `autoConfigure storageArray` command creates the disk groups on a storage array, the virtual disks in the disk groups, and the hot spares for the storage array. When you use the `autoConfigure storageArray` command, define the following parameters:

- Type of physical disks (Serial Attached SCSI [SAS])
- RAID level
- Number of physical disks in a disk group
- Number of disk groups
- Number of virtual disks in each disk group
- Number of hot spares
- Size of each segment on the physical disks

After defining these parameters, the MD Storage Manager automatically creates the disk groups, virtual disks, and hot spares. The RAID controller modules assign disk group and virtual disk numbers as they are created. After MD Storage Manager creates the initial configuration, you can use the `set virtualDisk` command to define virtual disk labels.

Before running the `autoConfigure storageArray` command, run the `show storageArray autoConfigure` command. The `show storageArray autoConfigure` command returns a list of parameter values that MD Storage Manager uses to create a storage array. Change any of the parameter values by entering new values for the parameters when you run the `autoConfigure storageArray` command. If you are satisfied with the parameter values that the `show storageArray autoConfiguration` command returns, run the `autoConfigure storageArray` command without new parameter values.

The following syntax is the general form of `autoConfigure storageArray` command:

```
autoConfigure storageArray [physicalDiskType=
(SAS) raidLevel=(0 | 1 | 5 | 6) | diskGroupWidth=
numberOfPhysicalDisks | diskGroupCount=
numberOfDiskGroups | virtualDisksPerGroupCount=
numberOfVirtualDisksPerGroup | hotSpareCount=
numberOfHotspares | segmentSize=segmentSizeValue]
```

NOTE: All parameters are optional. You can use one or all of the parameters as needed to define your configuration.

When you use the `autoConfigure storageArray` command without specifying the number of disk groups, the firmware determines how many virtual disks and disk groups to create. The firmware creates one disk group and one virtual disk up to the maximum number that the storage array can support. When you specify the number of disk groups, the firmware creates only that number of disk groups. When you create more than one disk group, all of the disk groups have the same number of physical disks and the same number of virtual disks.

- The `diskGroupWidth` parameter defines the number of unassigned physical disks wanted for each new disk group.
- The `diskGroupCount` parameter defines the number of new disk groups wanted in the storage array.
- The `virtualDisksPerGroupCount` parameter defines the number of virtual disks wanted in each disk group.
- The `hotSpareCount` parameter defines the number of hot spares wanted in each disk group.
- The `segmentSize` parameter defines the amount of data in kilobytes that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. The smallest units of storage are data blocks. Each data block stores 512 bytes of data. The size of a segment determines how many data blocks that it contains. An 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

  Valid values for the segment size are `8`, `16`, `32`, `64`, `128`, `256`, and `512`.

  When you enter a value for the segment size, the value is checked against the supported values provided by the RAID controller module at run time. If the value you enter is not valid, the RAID controller module returns a list of valid values.

If the virtual disk is for a single user with large I/O requests (such as multimedia), performance is maximized when a single I/O request can be serviced with a single data stripe. A data stripe is the segment size multiplied by the number of physical disks in the disk group that are used for data storage. In this environment, multiple physical disks are used for the same request, but each physical disk is accessed only once.

For optimal performance in a multi-user database or file system storage environment, set the segment size to minimize the number of physical disks needed to satisfy an I/O request. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests.

After you have finished creating the disk groups and virtual disks by using the `autoConfigure storageArray` command, you can further define the properties of the virtual disks in a configuration using the `set virtualDisk` command. (See Modifying Your Configuration.)

Example Of The Auto Configuration Command

```
client>smcli 123.45.67.89 -c "autoConfigure
storageArray physicalDiskType=SAS raidLevel=5
diskGroupWidth=8 diskGroupCount=3
virtualDisksPerGroupCount=4 hotSpareCount=2
segmentSize=8;"
```

The command in this example creates a storage array configuration that uses SAS physical disks set to RAID level 5. Three disk groups are created. Each disk group consists of eight physical disks configured into four virtual disks. The storage array has two hot spares, and segment size for each virtual disk is 8 KB.

Modifying Your Configuration

After creating your initial configuration, modify the properties of the configuration to ensure that it meets your requirements for data storage. Use the following commands to modify the properties of your configuration:

- `autoConfigure storageArray`
- `create virtualDisk`

Use the `set` commands to modify a storage array configuration. This section explains how to modify the following properties:

- Storage array password
- Simple Mail Transport Protocol (SMTP) and Simple Network Management Protocol (SNMP) alerts
- RAID controller module clocks
- Storage array host type
- Global hot spares

NOTE: Before modifying your configuration, save a copy of your current configuration to a file (see Saving A Configuration To A File). If you have problems with your modifications, you can use the information in the file to restore your previous configuration.

Setting The Storage Array Password

The `set storageArray` command enables you to define a password for a storage array. The following syntax is the general form of the command:

```
set storageArray password="password"
```

The `password` parameter defines a password for the storage array. Passwords provide added security to a storage array to reduce the possibility of implementing destructive commands.

CAUTION: Implementing destructive commands can cause serious damage, including data loss.

NOTE: CLI commands do not have interactive warnings for destructive commands.

Unless you define a password for the storage array, anyone can run all of the script commands. A password protects the storage array from any command that the RAID controller modules consider destructive. A destructive command is any command that can change the state of the storage array, such

as virtual disk creation, reset, delete, rename, or change. If you have more than one storage array in a storage configuration, each array has a separate password. Passwords can have a maximum length of 30 characters. You must put quotation marks (" ") around the password. The following example shows how to use the `set storageArray` command to define a password:

```
client>smcli 123.45.67.89 -c "set storageArray
password=\"1a2b3c4d5e"\;"
```

Setting Up SMTP And SNMP Alerts

The storage array can be set up to send automatic e-mail alert messages to specified email addresses when specific events occur. View the current alert configuration settings using the following command:

```
SMcli -d -i -s -w -v -S
```

By default, all alert configuration settings are **None**.

The following example shows how to set the mail server IP and the sender address configurations for SMTP alerts:

```
SMcli -m 123.45.67.892 -F
MyStorageArrayEvent@MyCompany.com
```

or

```
SMcli -m MyCompany.com -F
MyStorageArrayEvent@MyCompany.com
```

An example of a command to set the email alert destination and specify that only event information is to be sent is:

```
SMcli -a email:MyCompanySupport@MyCompany.com
123.45.67.89 -I eventOnly
```

The following example shows how to set the SNMP trap alert configuration. In this example, the trap destination is 123.45.67.891. The storage array is 123.45.67.892, and the community name is **public**.

```
SMcli -a trap:public, 123.45.67.891 123.45.67.892
```

Setting The RAID Controller Module Clocks

To synchronize the clocks on the RAID controller modules with the host, use the `set storageArray time` command. Running this command helps ensure that event timestamps written by RAID controller modules to the Major Event Log (MEL) match event timestamps written to the host log files. The RAID controller modules remain available during synchronization. An example of the command is:

```
client>smcli 123.45.67.89 -c "set storageArray time;"
```

Setting The Storage Array Host Type

The `set storageArray` command enables you to define the default host type. The following syntax is the general form of the command:

```
set storageArray defaultHostType=(hostTypeName | hostTypeIdentifier)
```

The `defaultHostType` parameter defines how the RAID controller modules communicate with the operating system on undefined hosts connected to the storage array. This parameter defines the host type only for storage array data I/O activities; it does not define the host type for the management station. The operating system can be Windows or Linux. For example, if you set the `defaultHostType` to Linux, the RAID controller module communicates with any undefined host if the undefined host is

running Linux. Typically, you need to change the host type only when you are setting up the storage array. The only time you might need to use this parameter is if you need to change how the storage array behaves relative to the hosts.

Before you can define the default host type, you need to determine what host types are connected to the storage array. To return information about host types connected to the storage array, you can use the `show storageArray` command with the `defaultHostType` parameter or `hostTypeTable` parameter. This command returns a list of the host types with which the RAID controller modules can communicate; it does not return a list of the hosts. The following examples show how to use the `defaultHostType` parameter and the `hostTypeTable` parameter:

```
client>smcli 123.45.67.89 -c "show storageArray defaultHostType;"
```

```
client>smcli 123.45.67.89 -c "show storageArray hostTypeTable;"
```

The following example shows how to define a specific default host type:

```
client>smcli 123.45.67.89 -c "set storageArray defaultHostType=11;"
```

The value 11 is the host type index value from the host type table.

Setting Modification Priority

Modification priority defines how much processing time is allocated for virtual disk modification operations. Time allocated for virtual disk modification operations affects system performance. Increases in virtual disk modification priority can reduce read/write performance. Operations affected by modification priority include:

- Copyback
- Reconstruction
- Initialization
- Changing segment size
- Defragmentation of a disk group
- Adding free capacity to a disk group
- Changing the RAID level of a disk group

The lowest priority rate favors system performance, but the modification operation takes longer. The highest priority rate favors the modification operation, but the system performance might be degraded.

The `set virtualDisk` command enables you to define the modification priority for a virtual disk. The following syntax is the general form of the command:

```
set (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNamen] | virtualDisk <wwid> |
accessVirtualDisk) modificationPriority=(highest |
high | medium | low | lowest)
```

The following example shows how to use this command to set the modification priority for virtual disks named **Engineering 1** and **Engineering 2**:

```
client>smcli 123.45.67.89 -c "set virtualDisks
[\"Engineering_1\" \"Engineering_2\"]
modificationPriority=lowest;"
```

The modification rate is set to `lowest` so that system performance is not significantly reduced by modification operations.

Assigning Global Hot Spares

Hot spare physical disks can replace any failed physical disk in the storage array. The hot spare must be the same type of physical disk as the physical disk that failed and must have capacity greater than or equal to any physical disk that can fail. If a hot spare is smaller than a failed physical disk, the hot spare cannot be used to rebuild the data from the failed physical disk. Hot spares are available only for RAID levels 1 or 5.

You can assign or unassign global hot spares by using the `set physicalDisk` command. To use this command, you must perform these steps:

1. Identify the location of the physical disks by enclosure ID and slot ID.
2. Set the `hotSpare` parameter to `TRUE` to enable the hot spare or `FALSE` to disable an existing hot spare.

The following syntax is the general form of the command:

```
set (physicalDisk [enclosureID,slotID] |
physicalDisks [enclosureID0,slotID0 ...
enclosureIDn,slotIDn] hotSpare=(TRUE | FALSE)
```

The following example shows how to use this command to set hot spare physical disks:

```
client>smcli 123.45.67.89 -c "set physicalDisks
[0,2 0,3] hotSpare=TRUE;"
```

Enter the enclosure ID and slot ID of each physical disk that you want to use. You must put brackets ([ ]) around the list. Separate the enclosure ID and slot ID of a physical disk by a comma. Separate each enclosure ID and slot ID pair by a space.

Selecting The Event Levels For Alert Notifications

The MD storage management software has four event levels: Critical, Informational, Warning, and Debug. You can configure the MD storage management software to send alert notifications for all of these event levels or only for certain event levels.

A background task called the persistent monitor runs independently of the MD storage management software and monitors the occurrence of events on all of the managed storage arrays. The persistent monitor is installed automatically with the MD storage management software. When an event occurs, alert notifications in the form of emails and SNMP trap messages are sent to the destination addresses that are specified in the **Configure Alerts** dialog. For more information about how to specify the destination addresses, refer to the *Configuring the Email and SNMP Alert Notification Settings* online help topic in the Enterprise Management Window (EMW).

When the persistent monitor starts for the first time, a properties file is created in the directory where the MD storage management software files are located. The properties file can be configured to enable or disable local logging in the Windows and UNIX operating systems. By default, local logging is enabled in the properties file.

When local logging is enabled in the Windows operating system, the persistent monitor logs the event information in the **Windows Event Log** file. When local logging is enabled in the UNIX operating system, the persistent monitor logs the event information in the **syslog** file. The properties file can also be configured to enable or disable remote **syslog** notification. You must restart the persistent monitor service after configuring the properties file for the changes to take effect.

You can configure the MD storage management software to notify only the event levels that you specify. For example, you can configure alert notifications only for Critical and Warning events.

1. Open the **Configure Alerts** dialog by performing one of these actions:
   a. Select a storage array in the **Devices** tab in the EMW.
   b. Select **Edit → Configure Alerts.**

      The **Configure Alerts** dialog is displayed. Go to step 3.
   c. Select **Configure Alerts** in the **Setup** tab in the EMW. Go to step 2.
2. Select the **All storage arrays** radio button, and click **OK**.

   The **Configure Alerts** dialog is displayed.
3. Select the **Filtering** tab.
4. Select the check boxes next to the event levels for which the MD storage management software must send alert notifications.

   NOTE: The configuration to send alert notifications for only certain event levels applies to all of the managed storage arrays in the MD storage management software.
5. Click **OK**.

Configuring Alert Notifications

You can use the **Configure Alerts** dialog box to set up an email alert notification in the event of an error on the network. Alert emails are sent to the specified global mail server and sender email addresses in the event of an error on the selected hosts of storage arrays. You can choose to be alerted to all problems or individual problems.

1. Open the **Configure Alerts** dialog box by performing one of these actions:
   a. Select a storage array in the **Devices** tab, and then select **Edit → Configure Alerts.**

      The **Configure Alerts** dialog box appears. Go to step 4.
   b. On the **Setup** tab, select **Edit → Configure Alerts.** Go to step 2.
2. Select one of the following radio buttons to specify an alert level:
   a. All storage arrays – To send an alert email about events on all storage arrays. Click **OK**.

      The **Configure Alerts** dialog box appears. Go to step 4.
   b. An individual storage array – To send an alert email about events that occur on only a specified storage array. Click **OK**.

      The **Select Storage Array** dialog box appears. Go to step 3.
3. Select the storage array for which you want to receive email alerts and click **OK**.

   The **Configure Alerts** dialog box appears.

   NOTE: If you do not know which storage array to select, click **Blink** to turn on the indicator lights of the storage array.
4. Fill in the information for the selected tab and click **OK**.

   The dialog box closes, and the Enterprise Management Window appears. An email alert is sent to the specified email address when an error occurs on the storage arrays or hosts that you selected.

5

Using The Snapshot Feature

The following types of virtual disk snapshot premium features are supported on the MD storage array:

- Snapshot Virtual Disks using multiple point-in-time (PiT) groups
- Snapshot Virtual Disks (Legacy) using a separate repository for each snapshot

NOTE: This section describes the Snapshot Virtual Disk using PiT groups. If you are using the Snapshot Virtual Disk (Legacy) premium feature, see Using The Snapshot (Legacy) Feature.

A snapshot image is a logical image of the content of an associated base virtual disk created at a specific point-in-time, often known as a restore point. This type of image is not directly readable or writable to a host since the snapshot image is used to save data from the base virtual disk only. To allow the host to access a copy of the data in a snapshot image, you must create a snapshot virtual disk. This snapshot virtual disk contains its own repository, which is used to save subsequent modifications made by the host application to the base virtual disk without affecting the referenced snapshot image.

Before Using Snapshot CLI Commands

There are two types of virtual disk snapshot premium features supported on your storage array. Depending on your RAID controller firmware version, you may be using the legacy version of the Snapshot feature. For more information, see Using The Snapshot (Legacy) Feature.

NOTE: Ensure that you know which type of snapshot premium feature you have activated on your storage array.

For information on the differences between the two snapshot features, see the *Administrator's Guide*.

Snapshot Images And Groups

A snapshot image is a logical image of the content of an associated base virtual disk created at a specific point-in-time, also called a restore point. This type of image is not directly readable or writable to a host since the snapshot image is used to save data from the base virtual disk only. To allow the host to access a copy of the data in a snapshot image, you must create a snapshot virtual disk. This snapshot virtual disk contains its own repository, which is used to save subsequent modifications made by the host application to the base virtual disk without affecting the referenced snapshot image.

A snapshot image differs from a snapshot (legacy) in the following ways:

- A snapshot image uses one repository for all snapshot images associated with a base virtual disk, improving performance when there are updates to the base virtual disk.
- A snapshot image only exists within a snapshot group. To make the snapshot image read/write accessible by hosts, you must create a snapshot virtual disk.

For more information on the Snapshot feature, see the *Administrator's Guide*.

Only the following can be included in a snapshot image:

- Standard virtual disks
- Thin provisioned virtual disks
- Consistency groups

Snapshot Groups And Snapshot Consistency Groups

The Snapshot Virtual Disk premium feature supports two types of snapshot groups:

- Snapshot groups
- Consistency groups

Snapshot Groups

The purpose of a snapshot group is to create a sequence of snapshot images on a given base virtual disk without impacting performance. You can set up a schedule for a snapshot group to automatically create a snapshot image at a specific time in the future or on a regular basis.

The following rules apply when creating a snapshot group:

- Snapshot groups can be created with or without snapshot images.
- Each snapshot image is a member of only one snapshot group.
- Standard virtual disks and thin virtual disks are the only types of virtual disks that can contain a snapshot group. Snapshot virtual disks cannot contain snapshot groups.
- The base virtual disk of a snapshot group can reside on either a disk group or a disk pool.
- Snapshot virtual disks and snapshot groups cannot exist on the same base virtual disk.
- A snapshot group uses a repository to save all data for the snapshot images contained in the group. A snapshot image operation uses less disk space than a full physical copy because the data stored in the repository is only the data that has changed since the latest snapshot image.
- A snapshot group is initially created with one repository virtual disk. The repository contains a small amount of data, which increases over time with subsequent data updates. You can increase the size of the repository by either increasing the capacity of the repository or adding virtual disks to the repository.

Snapshot Consistency Groups

To perform the same snapshot image operations on multiple virtual disks, you can create a consistency group containing the virtual disks. Any operation performed on the consistency group is performed simultaneously on all of the virtual disks in that group, which creates consistent copies of data between each virtual disk. Consistency groups are commonly used to create, schedule or rollback virtual disks.

Each virtual disk belonging to a consistency group is referred to as a member virtual disk. When you add a virtual disk to a consistency group, the system automatically creates a new snapshot group that corresponds to this member virtual disk. You can set up a schedule for a consistency group to automatically create a snapshot image of each member virtual disk at a specific time and/or on a regular basis.

This synchronized snapshot of all the virtual disks is especially suitable for applications that span multiple virtual disks, such as a database application containing log files on one virtual disk and the database itself on another.

The following rules apply when creating a consistency group:

- Consistency groups can be created initially with or without member virtual disks.
- Snapshot images can be created for a consistency group to enable consistent snapshot images between all member virtual disks.
- Consistency groups can be rolled back.
- A virtual disk can belong to multiple consistency groups.
- Only standard virtual disks and thin virtual disks can be included in a consistency group.
- Snapshots created using the Snapshot Virtual Disk (Legacy) premium feature cannot be included in a consistency group.
- A base virtual disk can reside on either a disk group or disk pool.

Understanding Snapshot Repositories

Repositories are system-created virtual disks used to hold write data for snapshots, snapshot groups and consistency groups. During creation of either group or write-enabled snapshot virtual disk, an associated repository is automatically created. By default, one individual repository virtual disk is created for each snapshot group or snapshot image. You can create the overall repository automatically using the default settings, or you can manually create the repository by defining specific capacity settings.

A snapshot virtual disk allows the host access to a copy of the data contained in a snapshot image. A snapshot image is not directly read or write accessible to the host and is used only to save data captured from the base virtual disk.

Snapshot Consistency Group Repositories

A snapshot consistency group is made up of simultaneous snapshots of multiple virtual disks. Each virtual disk that belongs to a consistency group is referred to as a member virtual disk. When you add a virtual disk to a consistency group, the system automatically creates a new snapshot group that corresponds to this member virtual disk. A consistency group repository is also created for each member virtual disk in a consistency group in order to save data for all snapshot images in the group.

A consistency group snapshot image comprises multiple snapshot virtual disks. Its purpose is to provide host access to a snapshot image that has been taken for each member virtual disk at the same moment in time. A consistency group snapshot image is not directly read or write accessible to hosts; it is used only to save the data captured from the base virtual disk. The consistency group snapshot virtual disk can be designated as either read-only or read-write. Read-write consistency group snapshot virtual disks require a repository for each member virtual disk in order to save any subsequent modifications made by the host application to the base virtual disk without affecting the referenced snapshot image. Each member repository is created when the consistency group snapshot virtual disk is created.

Consistency Groups And Remote Replication

Although a virtual disk can belong to multiple consistency groups, you must create separate consistency groups for snapshot images and Remote Replication.

When a base virtual disk containing a consistency group is added to Remote Replication (non-legacy, asynchronous), the repository will automatically purge the oldest snapshot image and set the auto-delete limit to the maximum allowable snapshot limit for a consistency group.

Additionally, all member virtual disks belonging to both a snapshot consistency group and a Remote Replication group must belong to the same Remote Replication group.

Creating Snapshot Images

Guidelines before creating a snapshot image:

- If you attempt to create a snapshot image on a snapshot group and that snapshot group has reached its maximum number of snapshot images. The `failBaseWrites` or `purgeSnapImages` parameters used with the `create snapGroup` command allows you to choose to either fail the write attempt or automatically purge a specified number of older snapshot images.

  ```
  create snapGroup
  userLabel="snapGroupName"
  sourceVirtualDisk="virtualDiskName"
  repositoryVirtualDisk="repos_XXXX"
  [repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
  rollbackPriority=(high | highest | medium | low | lowest)|
  repositoryFullLimit=percentValue |
  autoDeleteLimit=numberOfSnapImages]
  ```

- If you attempt to create a snapshot image and either of the two conditions below are present, the creation might fail:
  - The base virtual disk that contains this snapshot image is a member of a Remote Replication group.
  - The base virtual disk is currently synchronizing. When synchronization is complete, the snapshot image creation will complete.

Creating A Snapshot Image

Before creating a snapshot image, you must have at least one snapshot group into which you can add the snapshot image.

Create a snapshot image using the following command:

```
create snapImage (snapGroup="snapGroupName" | snapGroups=("snapGroupName1"...
"snapGroupName2 ..."))
```

where, `snapGroupName` or `snapGroupNames` is the name of the snapshot group (or groups) that you specify to hold the snapshot image.

NOTE: Ensure that you have either existing repositories, enough free capacity nodes, or available unconfigured capacity for the storage array on which you are creating the snapshot group repository.

Deleting A Snapshot Image

Use the `delete snapshot image` command to delete the oldest snapshot image from a snapshot group or consistency group.

After a snapshot image is deleted from a snapshot group:

- The snapshot image is deleted from the storage array.
- The repository's reserve space is released for reuse within the snapshot group.
- All snapshot virtual disks associated with deleted snapshot image are disabled.

Use the following command to delete a snapshot image from a consistency group:

```
delete snapImage (snapGroup="snapGroupName" | snapGroups=("snapGroupName1"...
"snapGroupNamen")) (deleteCount=numberOfSnapImages |
```

```
retainCount=numberOfSnapImages | ignoreSnapVirtualDisk=(TRUE | FALSE) |
snapImageID=OLDEST)
```

Creating A Consistency Group Snapshot Virtual Disk

Creating a snapshot virtual disk of a consistency group creates a viewable virtual disk of specific images in the consistency group. A consistency group snapshot virtual disk can be made up of a single base virtual disk or multiple base virtual disks from the consistency group.

This command also allows you to set the following attributes for the snapshot virtual disk:

- read-only
- repository full value
- automatic repository selection

Unique command syntax and naming rules apply. The name of a snapshot image has two parts separated by a colon (:):

- identifier of the snapshot group
- identifier of the snapshot image

If you do not specify the `repositoryVirtualDiskType` or `readOnly` parameters, the repositories for the consistency group snapshot virtual disk will be selected automatically. If the disk group or disk pool where the base virtual disk resides does not have enough space, the command will fail.

Create a read/write consistency group snapshot virtual disk on a snapshot consistency group named **snapCG1** with these three members **cgm1**, **cgm2**, and **cgm3**. The repository virtual disks already exist and are selected in this command.

```
create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk1"
cgSnapImageID="snapCG1:oldest" members=(cgm1:repos_0010 cgm2:repos_0011
cgm3:repos_0007);
```

Note the colon (:) in the name of the snapshot image to be included in the consistency group snapshot virtual disk. The colon is a delimiter that separates the name of the snapshot virtual disk from a particular snapshot image that you might want to use. Any of the following three options can be used following the colon:

- An integer value that is the actual sequence number of the snapshot image.
- `NEWEST` specifies the latest consistency group snapshot image.
- `OLDEST` specifies the earliest snapshot image created.

The use of the colon following the names of the members of the snapshot consistency group define the mapping between the member and a repository virtual disk. For example:

```
cgm1:repos_10, member cgm1 maps to repository virtual disk repos_0010
```

Create a read-only consistency group snapshot virtual disk on a snapshot consistency group named **snapCG1** with members **cgm1**, **cgm2**, and **cgm3** using the following command:

```
create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk3"
cgSnapImageID="snapCG1:oldest" readOnly;
```

Create a consistency group snapshot virtual disk that has a repository full limit set to 60 percent on a snapshot consistency group named **snapCG1** with these three members **cgm1**, **cgm2**, and **cgm3**, with the following command:

```
create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk3"
cgSnapImageID="snapCG1:oldest" repositoryFullLimit=60;
```

Create a read/write consistency group snapshot virtual disk with automatic repository selection on a snapshot consistency group named **snapCG1** with members **cgm1**, **cgm2**, and **cgm3**, with the following command:

```
create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk4"
cgSnapImageID="snapCG1:oldest";
```

Creating A Consistency Group Snapshot Image

To create a new snapshot image for each base virtual disk in snapshot consistency group, use the `create cgSnapImage` command. The command will suspend all pending I/O operations to each base virtual disk that is a member of the consistency group before creating the snapshot images. If the snapshot image cannot be completed successfully for all of the consistency group members, this command fails and new snapshot images are not created.

Since all members of a snapshot consistency group normally contain the same number of snapshot images, adding a new member to a snapshot consistency group with this command create a new member lacking previously created snapshot images. This is not an indication of an error condition.

To create a consistency group snapshot image:

```
create cgSnapImage consistencyGroup="consistencyGroupName"
```

where, `consistencyGroup` is the name of the consistency group for which you are creating snapshot images.

Deleting A Snapshot Virtual Disk Or A Consistency Group Snapshot Virtual Disk

You can use the following command to delete a snapshot virtual disk or a consistency group snapshot virtual disk. Optionally, you can also delete the repository members.

```
delete cgSnapVirtualDisk ["snapVirtualDiskName"] [deleteRepositoryMembers=(TRUE
| FALSE)];
```

where, `snapVirtualDiskName` is the snapshot you want to delete. The `deleteRepositoryMembers=TRUE` parameter will preserve the member virtual disks (default). Setting the parameter to `FALSE` will delete the member virtual disks.

Deleting A Consistency Group Snapshot Image

You can delete the snapshot images in a consistency group. However, when you delete a consistency group snapshot image that is associated with a consistency group snapshot virtual disk, the corresponding snapshot virtual disk member in the consistency group snapshot virtual disk is transitioned to a Stopped state. Being in this state means that the snapshot virtual disk member no longer has a relationship to the snapshot group of the deleted snapshot image. A snapshot virtual disk member in a Stopped state maintains its relationship to its consistency group snapshot virtual disk.

When a snapshot image(s) is deleted from a consistency group:

- The snapshot is deleted from the storage array.

- The repository's reserve space for reuse within the consistency group is released.
- Any member virtual disk associated with the deleted snapshot image(s) is moved to a Stopped state.
- The member snapshot virtual disks associated with the deleted snapshot image(s) is deleted.

To delete two snapshot images named **all_data_1** from a consistency group, use the following command:

```
delete cgSnapImage consistencyGroup="all_data_1"
```

To succeed, all snapshot images must be deleted or the command will fail and no snapshots will be deleted.

Scheduling Snapshot Images

Using the `enableSchedule` and `schedule` parameters with virtual disk and consistency group commands, you can schedule regular snapshot images to recover files or schedule regular backups. The schedule can be set when you initially create a snapshot group or consistency group, or you can add it later to an existing snapshot group or consistency group.

You can create a schedule that runs daily or weekly in which you select specific days of the week (Sunday through Saturday).

An example of the `set snapGroup` command using the schedule parameters:

```
set snapGroup ["snapGroupName"] enableSchedule=(TRUE | FALSE)
schedule=(immediate | snapshotSchedule)
```

Valid schedule values for the schedule parameter are `immediate`, `startDate`, `scheduleDay`, `startTime`, `scheduleInterval`, `endDate`, `noEndDate`, `timesPerDay` and `timeZone`.

For more information, see:

- Set Snapshot Group Attributes
- Create Snapshot Group
- Create Consistency Group

Starting, Stopping And Resuming A Snapshot Rollback

The command line interface allows you to:

- start a rollback operation from multiple snapshot images concurrently
- stop a rollback operation
- resume a rollback operation

When rolling back, the host will have immediate access to the new-rolled-back base virtual disk, but the existing base virtual disk will not allow the host read-write access once the rollback is initiated.

NOTE: Create a snapshot of the base virtual disk just prior to initiating the rollback to preserve the pre-rollback base virtual disk for recovery purposes.

When you start a rollback operation for a set of snapshot images, the content of the base virtual disk changes immediately to match the point-in-time content of the selected snapshot image virtual disk. The base virtual disk immediately becomes available for read/write requests after the rollback operation has successfully completed.

To start a rollback for a snapshot group named **snapGroup1**:

```
start snapImage ["snapGroup1"] rollback
```

You can specify a snapshot image rollback for a specific snapshot image in a snapshot group named **12345**:

```
stop snapImage ["snapGroup1:12345"] rollback;
```

To stop a rollback operation for the most recent snapshot image in a snapshot group that has the name **snapGroup1**:

```
stop snapImage ["snapGroup1:newest"] rollback;
```

The repository virtual disk associated with the snapshot image continues to track any new changes between the base virtual disk and the snapshot image virtual disk that occur after the rollback operation is completed.

To resume a rollback operation for the same snapshot image and snapshot group:

```
resume snapImage ["snapgroup1:12345"] rollback;
```

Creating A Snapshot Group

A snapshot group is a sequence of point-in-time images of a single associated base virtual disk. A snapshot group uses a repository to save data for all snapshot images contained in the group. The repository is created at the same time the snapshot group is created.

Guidelines when creating a snapshot group:

- When a base virtual disk that contains a snapshot group is added to an asynchronous remote replication group, the system automatically changes the repository full policy to automatically purge the oldest snapshot image and sets the auto-delete limit to the maximum allowable snapshot limit for a snapshot group.
- If the base virtual disk resides on a standard disk group, the repository members for any associated snapshot group can reside on either a standard disk group or a disk pool. If a base virtual disk resides on a disk pool, all repository members for any associated snapshot group must reside on the same disk pool as the base virtual disk.
- You cannot create a snapshot group on a failed virtual disk.
- If you attempt to create a snapshot image, that snapshot image creation operation might remain in a Pending state because of the following conditions:
  - The base virtual disk that contains this snapshot image is a member of an asynchronous remote replication group.
  - The base virtual disk is currently in a synchronizing operation. The snapshot image creation completes as soon as the synchronization operation is complete.

To create a snapshot group:

```
create snapGroup userLabel="snapGroupName" sourceVirtualDisk="virtualDiskName"
(repositoryVirtualDisk="repos_xxxx" | repositoryVirtualDisk=(diskGroupName
[capacity=capacityValue]) repositoryVirtualDisk=(diskPoolName
[capacity=capacityValue])) [repositoryFullPolicy=(failBaseWrites |
purgeSnapImages) | rollbackPriority=(highest | high | medium | low | lowest) |
repositoryFullLimit=percentValue | autoDeleteLimit=numberOfSnapImages |
enableSchedule=(TRUE | FALSE) schedule=(immediate | snapshotSchedule)]
```

Deleting A Snapshot Group

CAUTION: All of the data in the snapshot group is lost as soon as you run this command.

This command deletes an entire snapshot group and, if specified, the associated repository virtual disks.

```
delete snapGroup ["snapGroupName"] deleteRepositoryMembers=TRUE;
```

In the example, the repository virtual disks were preserved . By default, all member virtual disks in the repository virtual disk are retained as unused, unmapped standard virtual disks. Setting the `deleteRepositoryMembers` parameter to `FALSE` will delete the repository disks.

Reviving A Snapshot Group

If a snapshot group is not in a Failed state, it can be forced into an Optimal state. Running this command on a Failed snapshot will return an error message and the command will not complete.

```
revive snapGroup ["snapGroupName"];
```

where `snapGroupName` is the name of the snapshot group you want to force into an Optimal state.

Creating A Consistency Group

A consistency group contains simultaneous snapshots of multiple virtual disks to ensure consistent copies of a group of virtual disks. When you add a virtual disk to a consistency group, the system automatically creates a new snapshot group that corresponds to this member virtual disk.

The following command creates a new, empty consistency group. You must add the snapshot groups using the `set consistencyGroup addCGMember` command.

```
create consistencyGroup userLabel="consistencyGroupName"
[repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
repositoryFullLimit=percentValue | autoDeleteLimit=numberOfSnapImages |
enableSchedule=(TRUE | FALSE) | schedule (immediate | snapSchedule) |
rollbackPriority=(lowest | low | medium | high | highest)]
```

where `userLabel` is the name of the new consistency group you want to create. `repositoryFullPolicy` controls how you want snapshot processing to continue if the snapshot repository virtual disks are full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or delete the snapshot images (`purgeSnapImages`). The default action is to delete the snapshot images.

The `repositoryFullLimit` parameter controls the percentage of repository capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. `autoDeleteLimit` configures the automatic deletion thresholds of snapshot images to keep the total number of snapshot images in the snapshot group at or below a designated level. When this option is enabled, then any time a new snapshot image is created in the snapshot group, the system automatically deletes the oldest snapshot image in the group to comply with the limit value. This action frees repository capacity so it can be used to satisfy ongoing copy-on-write requirements for the remaining snapshot images.

For other values, see Create Consistency Group.

Deleting A Consistency Group

This command deletes a snapshot consistency group. You can either delete both the consistency group and the repository virtual disks contained by the consistency group, or you can delete only the consistency group and leave the repository virtual disks contained in the consistency group intact.

```
delete consistencyGroup [consistencyGroupName [deleteRepositoryMembers=(TRUE |
FALSE)]]
```

To delete the consistency group and the repository virtual disks, set the `deleteRepositoryMembers` parameter to `TRUE`. To retain the repository virtual disks, set this parameter to `FALSE`. The default setting is `FALSE`.

Once completed, this command:

- Deletes all existing snapshot images and snapshot virtual disks from the consistency group.
- Deletes all the associated snapshot images and snapshot virtual disks for each member virtual disk in the consistency group.
- Deletes all associated repositories for each member virtual disk in the consistency group (if selected).

Setting Consistency Group Attributes

To change or set the properties for a snapshot consistency group, use the `set consistencyGroup` command:

```
set consistencyGroup ["consistencyGroupName"] [userLabel="consistencyGroupName"
| repositoryFullPolicy=(failBaseWrites | purgeSnapImages)|
repositoryFullLimit=percentValue| autoDeleteLimit=numberOfSnapImages|
rollbackPriority=(lowest | low | medium | high | highest)]
```

The `repositoryFullPolicy` parameter determines how you want snapshot processing to continue if the snapshot repository virtual disks are full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or delete (`purgeSnapImages`) the snapshot images. The default action is to delete the images.

The `repositoryFullLimit` controls the percentage of repository capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. `autoDeleteLimit` configures the automatic deletion thresholds of snapshot images to keep the total number of snapshot images in the snapshot group at or below a designated level. When this option is enabled, then any time a new snapshot image is created in the snapshot group, the system automatically deletes the oldest snapshot image in the group to comply with the limit value. This action frees repository capacity so it can be used to satisfy ongoing copy-on-write requirements for the remaining snapshot images.

The `rollBackPriority` parameter sets priority level for a consistency group rollback in an operational storage array.

For example, use this command on a consistency group named **CGGroup_1**:

```
set consistencyGroup ["CGGroup_1"] autoDeleteLimit=6 rollbackPriority=low;
```

Valid values are `highest`, `high`, `medium`, `low`, or `lowest`. The rollback priority defines the amount of system resources that should be allocated to the rollback operation at the expense of system performance. A value of `high` indicates that the rollback operation is prioritized over all other host I/O. Lower values indicate that the rollback operation should be performed with minimal impact to host I/O.

`autoDeleteLimit` allows you to configure each snapshot group to perform automatic deletion of its snapshot images to keep the total number of snapshot images in the snapshot group at or below a maximum number of images. When the number of snapshot images in the snapshot group is at the maximum limit, the `autoDeleteLimit` parameter automatically deletes snapshot images whenever a new snapshot image is created in the snapshot group. The oldest snapshot images in the snapshot group until the maximum number of images defined with the parameter is met. Deleting snapshot images frees repository capacity so it can be used to satisfy ongoing copy-on-write requirements for the remaining snapshot images.

Adding A Member Virtual Disk To A Consistency Group

To add a new base virtual disk to an existing consistency group, specify an existing repository virtual disk for the new consistency group member or create a new repository virtual disk. When creating a new repository virtual disk, identify an existing disk group or disk pool in which to create the repository virtual disk.

To add a new base virtual disk to a consistency group with an existing virtual disk:

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName" repositoryVirtualDisk="repos_XXXX"
```

To add a new base virtual disk to a consistency group and create a new repository virtual disk using a disk group:

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName"
repositoryVirtualDisk=("diskGroupName" capacity=capacityValue(KB|MB|GB|TB|
bytes))
```

To add a new base virtual disk to a consistency group and create a new repository virtual disk using a disk pool:

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName"
repositoryVirtualDisk=("diskPoolName" capacity=capacityValue(KB|MB|GB|TB|bytes))
```

Restrictions

- The Snapshot premium feature must be enabled on the storage array.
- To add a new member virtual disk, the consistency group must fewer than the maximum number of virtual disks allowed in your configuration.
- If the base virtual disk resides on a standard disk group, the repository members for any associated consistency group can reside on either a standard disk group or a disk pool. If a base virtual disk resides on a disk pool, the repository members for any associated consistency group must reside on the same disk pool as the base virtual disk.
- You cannot add a virtual disk to a consistency group that is in a Failed state.

Removing A Member Virtual Disk From A Consistency Group

When removing a member virtual disk from an existing snapshot consistency group, you can also delete the repository virtual disk members from the consistency group, if desired.

The `remove` command syntax is shown below:

```
set consistencyGroup ["consistencyGroupName"]
removeCGMemberVirtualDisk="memberVirtualDiskName"
[deleteRepositoryMembers=(TRUE | FALSE)]
```

The `removeCGMemberVirtualDisk` parameter is the name of the member virtual disk that you want to remove; `deleteRepositoryMembers` determines whether the command removes all the repository members from the consistency group.

To remove a virtual disk names payroll-backup from a consistency group named **CGGroup_1**, but preserve the repository virtual disks:

```
set consistencyGroup ["CGGroup_1"] removeCGMemberVirtualDisk="payroll_backup"
deleteRepositoryMembers=FALSE;
```

To remove **CGGroup_1**, as well as all repository virtual disks:

```
set consistencyGroup ["CGGroup_1"] removeCGMemberVirtualDisk=" payroll_backup"
deleteRepositoryMembers=TRUE;
```

Or omit the `deleteRepositoryMembers` parameter, which invokes the default value of preserving repository virtual disks:

```
set consistencyGroup ["CGGroup_1"] removeCGMemberVirtualDisk=" payroll_backup";
```

Changing The Pre-read Consistency Check Setting Of An Overall Repository Virtual Disk

Use the Pre-Read Consistency Check option to define a storage array's capability to pre-read an overall repository virtual disk consistency information and determine whether the data of that overall repository virtual disk is consistent. An overall repository virtual disk that has this feature enabled returns read errors if the data is determined to be inconsistent by the RAID controller module firmware.

You can enable this option for overall repository virtual disks that contain consistency information. RAID Level 1, RAID Level 5, and RAID Level 6 maintain redundancy information.

You can change the Pre-Read Consistency Check for an overall repository for the following storage objects:

- Snapshot group
- Snapshot virtual disk
- Consistency group member virtual disk
- Replicated Pair

This command defines the properties for a virtual disk. You can use most parameters to define properties for one or more virtual disks. You can also use some parameters to define properties for only one virtual disk. The syntax definitions are separated to show which parameters apply to several virtual disks and which apply to only one virtual disk.

NOTE: In configurations where disk groups consist of more than 32 virtual disks, the operation can result in host I/O errors or internal RAID controller module reboots due to the expiration of the timeout period before the operation completes. If you experience host I/O errors or internal RAID controller module reboots, quiesce the host I/O and try the operation again.

Syntax Applicable To One Or More Virtual Disks

```
set (allVirtualDisks | virtualDisk ["virtualDiskName"] | virtualDisks
["virtualDiskName1" ... "virtualDiskNameN"] | virtualDisk <wwID>)
cacheFlushModifier=cacheFlushModifierValue cacheWithoutBatteryEnabled=(TRUE |
FALSE)
mediaScanEnabled=(TRUE | FALSE)
replicationCacheEnabled=(TRUE | FALSE)
modificationPriority=(highest | high | medium | low | lowest)
owner=(0 | 1)
```

```
preReadConsistencyCheck=(TRUE | FALSE)
readCacheEnabled=(TRUE | FALSE)
writeCacheEnabled=(TRUE | FALSE)
cacheReadPrefetch=(TRUE | FALSE)
```

Syntax Applicable To Only One Virtual Disk

```
set (virtualDisk ["virtualDiskName"] | virtualDisk <wwID>)
addCapacity=virtualDiskCapacity
[addPhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)]
consistencyCheckEnabled=(TRUE | FALSE) segmentSize=segmentSizeValue
userLabel=virtualDiskName
preReadConsistencyCheck=(TRUE | FALSE)
```

preReadConsistencyCheck

The setting to turn on or turn off preread consistency checking. Turning on preread consistency checking verifies the consistency of RAID redundancy data for the stripes containing the read data. Preread consistency checking is performed on read operations only. To turn on preread consistency checking, set this parameter to `TRUE`. To turn off preread consistency checking, set this parameter to `FALSE`.

NOTE: Do not use this parameter on non-redundant virtual disks, such as RAID 0 virtual disks.

Guidelines while using the Pre-Read Consistency Check option:

- Changing the Pre-Read Consistency Check setting modifies the setting only for the overall repository that you selected.
- The Pre-Read Consistency Check setting is applied to all individual repository virtual disks contained within the overall repository.
- If an overall repository virtual disk that is configured with pre-read is migrated to a RAID level that does not maintain consistency information, the metadata of the overall repository virtual disk continues to show that pre-read is enabled. However, reads to that overall repository virtual disk ignores consistency pre-read. If the virtual disk is subsequently migrated back to a RAID level that supports consistency, the option becomes available again.

NOTE: Enabling the option on overall repository virtual disks without consistency does not affect the virtual disk. However, the attribute is retained for that overall repository virtual disk if it is ever changed to one with consistency information.

Setting Snapshot Virtual Disk Repository Virtual Disk Capacity

To increase or decreases the capacity of a snapshot virtual disk repository virtual disk, use the `set snapVirtualDisk` command.

A snapshot repository virtual disk is an expandable virtual disk that is structured as a concatenated collection of up to 16 standard virtual disk entities. Initially, an expandable repository virtual disk has only a single element. The capacity of the expandable repository virtual disk is exactly that of the single element. You can increase the capacity of an expandable repository virtual disk by attaching additional standard virtual disks to it. The composite expandable repository virtual disk capacity then becomes the sum of the capacities of all of the concatenated standard virtual disks.

A snapshot group repository virtual disk must satisfy a minimum capacity requirement that is the sum of the following:

- 32 MB to support fixed overhead for the snapshot group and for copy-on-write processing.

- Capacity for rollback processing, which is 1/5000th of the capacity of the base virtual disk.

If you are creating a new repository virtual disk when you run this command, you must enter the name of a disk group or disk pool from which you want the repository virtual disk. Optionally, you can also define the capacity of the repository virtual disk. The following capacity values are supported:

- A percentage (integer value) representing an amount of the base virtual disk capacity.
- A decimal fraction representing a percentage of the base virtual disk capacity.
- A specific size for the repository virtual disk, in units of bytes, KB, MB, GB, or TB.

To increase capacity for a snapshot virtual disk, use the `increaseRepository Capacity` and `capacity` parameter as shown below:

```
set snapVirtualDisk[“snapVirtualDiskName”] increaseRepositoryCapacity
repositoryVirtualDisks=("repos_xxxx" | repositoryVirtualDisks=(diskGroupName
[capacity=capacityValue]) repositoryVirtualDisks=(diskPoolName
[capacity=capacityValue]))
```

To decrease capacity, replace `increaseRepositoryCapacity` with `decreaseRepository`:

```
set snapVirtualDisk ["snapVirtualDiskName"] decreaseRepositoryCapacity
count=numberOfVirtualDisks
```

Setting Snapshot Group Repository Virtual Disk Capacity

To increase or decrease capacity of a snapshot group repository virtual disk, use the same basic command syntax as shown previously for the virtual disk repository, except use the `set snapGroup` command and supply the snapshot group name:

```
set snapGroup ["snapGroupName"] increaseRepositoryCapacity
(repositoryVirtualDisks="repos_xxxx" | repositoryVirtualDisks=((diskGroupName
[capacity=capacityValue])) | repositoryVirtualDisks=((diskPoolName
[capacity=capacityValue])))
```

For more information about expanding storage objects, see the *Administrator's Guide*.

Reviving Disk Groups, Physical Disks, Snapshot Groups, And Snapshot Virtual Disks

The `revive` command forces failed physical disks, disk groups, snapshot groups and snapshot virtual disks into an Optimal state. However, this command should only be performed by qualified storage administrators.

CAUTION: Possible loss of data access—Correct use of this command depends on the data configuration on the physical disks in the disk group. Do not attempt to revive a physical disk or disk group unless you are explicitly directed by your Technical Support representative.

To revive a physical disk:

```
revive physicalDisk [enclosureID,drawerID,slotID]
```

For MD-series dense storage arrays, all three location attributes (*enclosureID*, *drawerID* and *slotID*) are required. For non-dense storage arrays, only *enclosureID* and *slotID* are required. Valid *enclosureID* values are `0` to `99`, *drawerID* values are `0` to `4` and *slotID* values are `0` to `31`.

To revive disk group:

```
revive diskGroup [diskGroupName]
```

To revive a snapshot virtual disk, the snapshot virtual disk must be a:

- standalone snapshot virtual disk
- snapshot virtual disk that is a member of a consistency group

If the snapshot virtual disk is not in a Failed state, the firmware displays an error message and does not run this command.

```
revive snapVirtualDisk ["snapVirtualDiskName"]
```

NOTE: This command for a snapshot virtual disk used in online virtual disk copy.

To revive a snapshot group:

```
revive snapGroup ["snapGroupName"]
```

6

Using The Snapshot (Legacy) Feature

The following types of virtual disk snapshot premium features are supported on the MD storage array:

- Snapshot Virtual Disks using multiple point-in-time (PiT) groups
- Snapshot Virtual Disks (Legacy) using a separate repository for each snapshot

NOTE: This section describes the Snapshot Virtual Disk (legacy) premium feature. If you are using the Snapshot Virtual Disk using PiT groups, see Using The Snapshot Feature.

This chapter describes how the Snapshot (legacy) feature works, lists the snapshot script commands, and explains how to use the commands to create snapshot virtual disks. Additional information about the Snapshot (legacy) feature and related definitions is available in the online help, the *Deployment Guide*, the *MD Storage Manager online help*, and the *Owner's Manual*.

The Snapshot (legacy) feature creates a snapshot virtual disk that you can use as a backup of your data. A snapshot virtual disk is a logical point-in-time image of a standard virtual disk. Because it is not a physical copy, a snapshot virtual disk is created more quickly than a physical copy and requires less physical disk space. Typically, you create a snapshot virtual disk so that an application, such as a backup application, can access the snapshot virtual disk. The application reads the data while the source virtual disk remains online and user accessible. You can also create several snapshot virtual disks of a source virtual disk and write data to the snapshot virtual disks to perform testing and analysis.

NOTE: If you ordered Premium Features for the Snapshot Virtual Disks, you would have received a Premium Features Activation card shipped in the same box as your Dell PowerVault MD storage array. Follow the directions on the card to obtain a key file and to enable the feature. For more information, see Premium Feature — Snapshot Virtual Disks in the *Owner's Manual*.

Snapshot virtual disks allow you to perform the following tasks:

- Create a complete image of the data on a source virtual disk at a particular point in time.
- Use only a small amount of disk space.
- Provide quick, frequent, nondisruptive backups; or test new versions of a database system without affecting actual data.
- Provide for snapshot virtual disks to be read, written, and copied.
- Use the same availability characteristics of the source virtual disk (such as redundant array of independent disks (RAID) protection and redundant path failover).
- Map the snapshot virtual disk and make it accessible to any host on a storage area network. You can make snapshot data available to secondary hosts for read and write access by mapping the snapshot to the hosts.
- Create up to four snapshots per virtual disk.

  NOTE: The maximum number of snapshot virtual disks is one-half of the total number of virtual disks supported by the RAID controller module.
- Increase the capacity of a snapshot virtual disk.

The following table lists the components that comprise a snapshot virtual disk and briefly describes what they do.

Table 9. Snapshot Virtual Disk Components

Component	Description
Source virtual disk	Standard virtual disk from which the snapshot is created
Snapshot virtual disk	Point-in-time image of a standard virtual disk
Snapshot repository virtual disk	Virtual disk that contains snapshot metadata and copy-on-write data for a particular snapshot virtual disk

The following table lists the snapshot virtual disk commands and brief descriptions of what the commands do.

Table 10. Snapshot Virtual Disk Commands

Command	Description
`create snapshotVirtualDisk`	Creates a snapshot virtual disk.
`re-create snapshot`	Starts a fresh copy-on-write operation by using an existing snapshot virtual disk.
`set (snapshotVirtualDisk)`	Defines the properties for a snapshot virtual disk and enables you to rename a snapshot virtual disk.
`stop snapshot`	Stops a copy-on-write operation.

Using Host Servers To Create An Initial Snapshot Virtual Disk

CAUTION: Before using the Snapshot Virtual Disks Premium Feature in a Microsoft Windows clustered configuration, you must first map the snapshot virtual disk to the cluster node that owns the source virtual disk. This ensures that the cluster nodes correctly recognize the snapshot virtual disk.

If you map the snapshot virtual disk to the node that does not own the source virtual disk before the snapshot enabling process is completed, the operating system may fail to correctly identify the snapshot virtual disk. This can result in data loss on the source virtual disk or an inaccessible snapshot.

NOTE: You can create concurrent snapshots of a source virtual disk on both the source disk group and on another disk group.

Before creating a Snapshot Virtual Disk, note the following:

- The following types of virtual disks are not valid source virtual disks: snapshot repository virtual disks, snapshot virtual disks, target virtual disks that are participating in a virtual disk copy.
- You cannot create a snapshot of a virtual disk that contains unreadable sectors.

- You must satisfy the requirements of your host operating system for creating snapshot virtual disks. Failure to meet the requirements of your host operating system results in an inaccurate point-in-time image of the source virtual disk or the target virtual disk in a virtual disk copy.

Creating A Snapshot Virtual Disk

The `create snapshotVirtualDisk` command provides three methods for defining the physical disks for your snapshot repository virtual disk:

- Define each physical disk for the snapshot repository virtual disk by enclosure ID and slot ID.
- Define a disk group in which the snapshot repository virtual disk resides. Optionally define the capacity of the repository virtual disk.
- Define the number of physical disks, but not specific physical disks, for the repository virtual disk.

When using the `create snapshotVirtualDisk` command to create a snapshot virtual disk, the standard virtual disk name for the source virtual disk is the minimum information required. When you provide only the standard virtual disk name, the storage management software provides default values for the other required property parameters for a snapshot virtual disk.

NOTE: In some cases, depending on the host operating system and any virtual disk manager software in use, the software prevents you from mapping the same host to both a source virtual disk and its associated snapshot virtual disk.

An error message appears in the command line when the utility cannot distinguish between the following:

- Source virtual disk and snapshot virtual disk (for example, if the snapshot virtual disk has been removed)
- Standard virtual disk and virtual disk copy (for example, if the virtual disk copy has been removed)

If you are running a Linux operating system, run the **hot_add** utility to register the snapshot virtual disk with the host operating system.

NOTE: The **hot_add** utility is not available for Windows.

Enabling The Snapshot Virtual Disk Feature

The first step in creating a snapshot virtual disk is to make sure the feature is enabled on the storage array. You need a feature key to enable the feature. The command for enabling the feature key file is:

`enable storageArray feature file="`*`filename`*`"`

where, the `file` parameter is the complete file path and file name of a valid feature key file. Enclose the file path and file name in quotation marks (" "). Valid file names for feature key files usually end with **.key** extension.

Creating A Snapshot Virtual Disk With User-Assigned Physical Disks

Creating a snapshot virtual disk by assigning the physical disks allows you to choose from the available physical disks when defining your storage array configuration. When you choose the physical disks for your snapshot virtual disk, you automatically create a new disk group. You can specify which physical disks to use and the RAID level for the new disk group.

Preparing Host Servers To Create An Initial Snapshot Virtual Disk

CAUTION: Before you create a new point-in-time image of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk(s) in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the Snapshot.

Before creating a snapshot virtual disk, the server has to be in the proper state. To ensure that the host server is properly prepared to create a snapshot virtual disk, you can either use an application to carry out this task, or you can perform the following steps:

1. Stop all I/O activity to the source.
2. Using your Windows system, flush the cache to the source. At the host prompt, type `SMrepassist -f <filename-identifier>` and press **<Enter>**.

   See "SMrepassist Utility" in the Owner's Manual for more information.
3. Remove the drive letter(s) of the source in Windows or unmount the virtual drive(s) in Linux to help guarantee a stable copy of the drive for the Snapshot. If this is not done, the snapshot operation reports that it has completed successfully, but the snapshot data is not updated properly.

   NOTE: Verify that the virtual disk has a status of Optimal or Disabled by clicking the **Summary** tab and then clicking the **Disk Groups & Virtual Disks** link.
4. Follow any additional instructions for your operating system. Failure to follow these additional instructions can create unusable snapshot virtual disks.

   NOTE: If your operating system requires additional instructions, you can find those instructions in your operating system documentation.

If you want to use a snapshot regularly, such as for backups, use the Disable Snapshot and Re-create Snapshot options to reuse the snapshot. Disabling and re-creating snapshots preserves the existing virtual disk-to-host mappings to the snapshot virtual disk.

After your server has been prepared, see Creating The Initial Snapshot Virtual Disk.

Creating The Initial Snapshot Virtual Disk

After preparing the host server(s) as specified in the preceding procedure, use the following examples to make a virtual disk snapshot.

The following syntax is the general form of the command to create a snapshot virtual disk:

```
create snapshotVirtualDisk sourceVirtualDisk=
"sourceVirtualDiskName" [repositoryRAIDLevel=(0 |
1 | 5) (repositoryPhysicalDisks=
(enclosureID0,slotID0 ... enclosureIDn,slotIDn)
userLabel="snapshotVirtualDiskName"
warningThresholdPercent=percentValue
repositoryPercentOfSource=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failSourceWrites |
failSnapShot)] [enclosureLossProtect=(TRUE |
FALSE)]enableSchedule=(TRUE | FALSE) | schedule=
(immediate | snapshotSchedule)]
```

NOTE: Use one or all of the optional parameters as needed to help define your configuration. You do not need to use any optional parameters.

See step 1 through step 4 in the preceding section, Preparing Host Servers To Create An Initial Snapshot Virtual Disk. The following example shows a command in which users assign the physical disks:

```
client>smcli 123.45.67.89 -c "create
snapshotVirtualDisk sourceVirtualDisk=
\"Mars_Spirit_4\" repositoryRAIDLevel=5
repositoryPhysicalDisks=(1,1 1,2 1,3 1,4 1,5);"
```

The command in this example creates a new snapshot of the source virtual disk **Mars_Spirit_4**. The snapshot repository virtual disk consists of five physical disks that form a new disk group. The new disk group has a RAID level of 5. This command also takes a snapshot of the source virtual disk, starting the copy-on-write operation.

See step 1 through step 4 in the preceding section, Preparing Host Servers To Create An Initial Snapshot Virtual Disk. The following example is the script file version of the command:

```
create snapshotVirtualDisk sourceVirtualDisk=
"Mars_Spirit_4" repositoryRAIDLevel=5
repositoryPhysicalDisks=(1,1 1,2 1,3 1,4 1,5);
```

A minimal version of this command might look like the following example:

```
client>smcli 123.45.67.89 -c "create
snapshotVirtualDisk sourceVirtualDisk=
\"Mars_Spirit_4\";"
```

The command in this example creates a new snapshot for the source virtual disk **Mars_Spirit_4**. The repository virtual disk is created in the same disk group as the source virtual disk, which means that the repository virtual disk has the same RAID level as the source virtual disk. This command starts the copy-on-write operation.

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Create An Initial Snapshot Virtual Disk. The following example is the script file version of the command:

```
create snapshotVirtualDisk sourceVirtualDisk= "Mars_Spirit_4";
```

Creating A Snapshot Virtual Disk With Software-Assigned Physical Disks

This version of the `create snapshotVirtualDisk` command lets you choose an existing disk group to place the snapshot repository virtual disk. The storage management software determines which physical disks to use. You can also define how much space to assign to the repository virtual disk. Because you are using an existing disk group, the RAID level for the snapshot virtual disk defaults to the RAID level of the disk group in which you place it. You cannot define the RAID level for the snapshot virtual disk. The general syntax for this command is:

```
create snapshotVirtualDisk sourceVirtualDisk=
"sourceVirtualDiskName" [repositoryDiskGroup=
diskGroupNumber freeCapacityArea=
freeCapacityIndexNumber userLabel=
"snapshotVirtualDiskName"
warningThresholdPercent=percentValue
repositoryPercentOfSource=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failSourceWrites |
failSnapShot)] [enclosureLossProtect=(TRUE |
FALSE)]
```

NOTE: Use one or all of the optional parameters as needed to define your configuration. It is not necessary to use any optional parameters.

The following example is a command in which software assigns the physical disks:

```
client>smcli 123.45.67.89 -c "create
snapshotVirtualDisk sourceVirtualDisk=
\"Mars_Spirit_4\" repositoryDiskGroup=2
freeCapacityArea=2;"
```

The command in this example creates a new snapshot repository virtual disk in disk group 2. The source virtual disk is **Mars_Spirit_4**. The size of the snapshot repository is 4 GB. This command also takes a snapshot of the source virtual disk, which starts the copy-on-write operation.

Define the capacity of a snapshot repository virtual disk as any percentage of the size of the source virtual disk. A value of 20 percent is an optimum number. In the previous example, the size of the snapshot repository is set to 4 GB. The underlying assumption is that the source virtual disk size is 20 GB (0.2 x 20 GB = 4 GB).

The following example is the script file version of the command:

```
create snapshotVirtualDisk sourceVirtualDisk=
"Mars_Spirit_4" repositoryDiskGroup=2
freeCapacityArea=2;
```

Creating A Snapshot Virtual Disk By Specifying A Number Of Physical Disks

With this version of the `create snapshotVirtualDisk` command, you must specify the number of physical disks and the RAID level for the snapshot repository virtual disk. This version of the `create snapshotVirtualDisk` command creates a new disk group. You must have physical disks in the storage array that are not assigned to a disk group for this command to work:

```
create snapshotVirtualDisk sourceVirtualDisk=
"sourceVirtualDiskName" [repositoryRAIDLevel=(0 |
1 | 5 | 6) repositoryPhysicalDiskCount=
numberOfPhysicalDisks physicalDiskType=(SAS)
userLabel="snapshotVirtualDiskName"
warningThresholdPercent=percentValue
repositoryPercentOfSource=percentValue
repositoryUserLabel="repositoryName"
repositoryFullPolicy=(failSourceWrites |
failSnapShot)] [enclosureLossProtect=(TRUE |
FALSE)]
```

NOTE: Use one or all optional parameters as needed to define your configuration. It is not necessary to use any optional parameters.

The following example is a command in which users specify the number of physical disks:

```
client>smcli 123.45.67.89 -c "create
snapshotVirtualDisk sourceVirtualDisk=
\"Mars_Spirit_4\" repositoryRAIDLevel=5
repositoryPhysicalDiskCount=3;"
```

The command in this example creates a new snapshot repository virtual disk that consists of three physical disks. The three physical disks comprise a new disk group with a RAID level of 5. This command also takes a snapshot of the source virtual disk, which starts the copy-on-write operation.

The following example is the script file version of the command:

```
create snapshotVirtualDisk sourceVirtualDisk=
"Mars_Spirit_4" repositoryRAIDLevel=5
repositoryPhysicalDiskCount=3;
```

User-Defined Parameters

Parameters for the `create snapshotVirtualDisk` command enable you to define the snapshot virtual disk to suit the requirements of your storage array. The following table lists the parameters and descriptions of Snapshot Virtual Disk.

Table 11. Snapshot Virtual Disk Parameters

Parameter	Description
`physicalDiskType`	Specifies the type of physical disk to use for the snapshot repository virtual disk. The type must be specified as Serial Attached SCSI (SAS). This parameter works only with the count-based repository method of defining a snapshot virtual disk.
`repositoryDiskGroup`	Specifies the disk group in which to build the snapshot virtual disk. Default builds the snapshot repository virtual disk in the same disk group as the source virtual disk.
`freeCapacityArea`	Specifies the amount of storage space to use for the snapshot repository virtual disk. Free storage space is defined in units of bytes, kilobytes, megabytes, or gigabytes.
`userLabel`	Specifies the name to give to the snapshot virtual disk. If you do not choose a name for the snapshot virtual disk, the RAID controller modules create a default name using the source virtual disk name. For example, if the source virtual disk name is **Mars_Spirit_4** and it does not have a snapshot virtual disk, the default snapshot virtual disk name is **Mars_Spirit_4-1**. If the source virtual disk already has *n* – 1 number of snapshot virtual disks, the default name is **Mars_Spirit_4-**n.
`repositoryUserLabel`	Specifies the name to give to the snapshot repository virtual disk. If you do not choose a name for the snapshot repository virtual disk, the RAID controller modules create a default name using the source virtual disk name. For example, if the source virtual disk name is **Mars_Spirit_4** and it does not have an associated snapshot repository virtual disk, the default snapshot repository virtual disk name is **Mars_Spirit_4-R1**. If the source virtual disk already has *n* – 1 number of snapshot repository virtual disks, the default name is **Mars_Spirit_4-Rn**.

Parameter	Description
warningThresholdPercent	Specifies how full to allow the snapshot repository virtual disk to get before sending a warning that the snapshot repository virtual disk is close to capacity. The warning value is a percentage of the total capacity of the snapshot repository virtual disk. The default value is 50, which represents 50 percent of total capacity. (Change this value using the `set snapshotVirtualDisk` command.)
repositoryPercentOfSource	Specifies the size of the snapshot repository virtual disk as a percentage of the source virtual disk size. The default value is 20, which represents 20 percent of the source virtual disk size.
repositoryFullPolicy	Specifies how snapshot processing continues if the snapshot repository virtual disk is full. You can choose to fail writes to the source virtual disk (`failSourceWrites`) or fail writes to the snapshot virtual disk (`failSnapShot`). The default value is `failSnapShot`.
enableSchedule	Turns on or off the ability to schedule a snapshot operation. To turn on snapshot scheduling, set the parameter to `TRUE`. To turn off snapshot scheduling, set the parameter to `FALSE`. Schedules a snapshot operation of the following type: • `immediate` • `startDate` • `scheduleDay` • `startTime` • `scheduleInterval` • `endDate` • `noEndDate` • `timesPerDay` • `timeZone`

The following example of the `create sourceVirtualDisk` command includes user-defined parameters:

```
client>smcli 123.45.67.89 -c "create
snapshotVirtualDisk sourceVirtualDisk=
\"Mars_Spirit_4\" repositoryRAIDLevel=5
repositoryPhysicalDiskCount=5 physicalDiskType=
SAS userLabel=\"Mars_Spirit_4_snap1\"
repositoryUserLabel=\"Mars_Spirit_4_rep1\"
warningThresholdPercent=75
repositoryPercentOfSource=40
repositoryFullPolicy=
failSnapShot;"enableSchedule=TRUE schedule=
(immediate | snapshotSchedule)]
```

The following example is the script file version of the command:

```
create snapshotVirtualDisk sourceVirtualDisk=
"Mars_Spirit_4" repositoryRAIDLevel=5
repositoryPhysicalDiskCount=5 physicalDiskType=
SAS userLabel="Mars_Spirit_4_snap1"
repositoryUserLabel="Mars_Spirit_4_rep1"
warningThresholdPercent=75
repositoryPercentOfSource=40
repositoryFullPolicy=failSnapShot;
```

NOTE: In the previous examples, the names for the snapshot virtual disk and repository virtual disk are defined by the user. If you do not choose to create names for the snapshot virtual disks or the repository virtual disks, the RAID controller modules provide default names. (See Names Of Snapshot Virtual Disks And Repository Virtual Disks for an explanation of naming conventions.)

Names Of Snapshot Virtual Disks And Repository Virtual Disks

The names of snapshot virtual disks and repository virtual disks can be any combination of alphanumeric characters, hyphens, and underscores. The maximum length of the virtual disk names is 30 characters. You must enclose the name in quotation marks. The character string cannot contain a new line. Make sure that you use unique names or the RAID controller module firmware returns an error.

One technique for naming the snapshot virtual disk and the repository virtual disk is to add a hyphenated suffix to the original name of the source virtual disk. The suffix distinguishes between the snapshot virtual disk and the repository virtual disk. For example, if you have a source virtual disk with a name **Engineering Data**, the snapshot virtual disk can have a name **Engineering Data-S1**. The repository virtual disk can have a name of **Engineering Data-R1**.

If you do not choose a unique name for either the snapshot virtual disk or repository virtual disk, the RAID controller modules create a default name by using the name of the source virtual disk. For example, if the name of the source virtual disk is **aaa** and it does not have a snapshot virtual disk, then the default name is **aaa-1**. If the source virtual disk already has $n - 1$ number of snapshot virtual disks, then the default name is **aaa-n**. Similarly, if the name of the source virtual disk is **aaa** and it does not have a repository virtual disk, then the default repository virtual disk name is **aaa-R1**. If the source virtual disk already has $n - 1$ number of repository virtual disks, then the default name is **aaa-R** *n*.

In the examples from the previous section, the user-defined name of the snapshot virtual disk was **Mars_Spirit_4_snap1**. The user-defined name of the repository virtual disk was **Mars_Spirit_4_rep1**. The default name provided by the RAID controller module for the snapshot virtual disk would be **Mars_Spirit_4-1**. The default name provided by the RAID controller module for the repository virtual disk would be **Mars_Spirit_4-R1**.

Changing Snapshot Virtual Disk Settings

The `set (snapshot) virtualDisk` command enables you to change the property settings for a snapshot virtual disk. Using this command, you can change the following parameters:

- Name of the snapshot virtual disk
- Warning threshold percent
- Repository full policy

The following example shows the command to change the name of a snapshot virtual disk:

```
client>smcli 123.45.67.89 -c "set virtualDisk
[\"Mars_Spirit_4-1\"] userLabel=\"Mars_Odyssey_3-2\";"
```

The following example is the script file version of the command:

```
set virtualDisk ["Mars_Spirit_4-1"] userLabel=
"Mars_Odyssey_3-2";
```

When you change the warning threshold percent and repository full policy, you can apply the changes to one or several snapshot virtual disks. The following example uses the `set (snapshot) virtualDisk` command to change these properties on more than one snapshot virtual disk:

```
client>smcli 123.45.67.89 -c "set virtualDisks
[\"Mars_Spirit_4-1\" \"Mars_Spirit_4-2\"
\"Mars_Spirit_4-3\"] warningThresholdPercent=50
repositoryFullPolicy=failSourceWrites;"
```

The following example is the script file version of the command:

```
set virtualDisks ["Mars_Spirit_4-1"
"Mars_Spirit_4-2" "Mars_Spirit_4-3"]
warningThresholdPercent=50 repositoryFullPolicy=
failSourceWrites;
```

Stopping And Deleting A Snapshot Virtual Disk

When you create a snapshot virtual disk, copy-on-write immediately starts running. As long as a snapshot virtual disk is enabled, storage array performance is affected by the copy-on-write operations to the associated snapshot repository virtual disk. If you no longer want copy-on-write operations to run, you can use the `stop snapshot virtualDisk` command to stop the copy-on-write operations. When you stop a snapshot virtual disk, the snapshot virtual disk and the repository virtual disk are still defined for the source virtual disk; only copy-on-write has stopped. The following example stops a snapshot virtual disk:

```
client>smcli 123.45.67.89 -c "stop snapshot
virtualDisks [\"Mars_Spirit_4-2\" \"Mars_Spirit_4-3\"];"
```

The following example is the script file version of the command:

```
stop snapshot virtualDisks ["Mars_Spirit_4-2"
"Mars_Spirit_4-3"];
```

When you stop the copy-on-write operations for a specific snapshot virtual disk, only that snapshot virtual disk is disabled. All other snapshot virtual disks remain in operation.

Re-creating The Snapshot Virtual Disk

To restart a copy-on-write operation, use the `recreate snapshot virtualDisk` command. This command starts a fresh copy-on-write operation using an existing snapshot virtual disk. When you restart a snapshot virtual disk, the snapshot virtual disk must have either an Optimal or a Disabled state.
The following conditions then occur:

- All copy-on-write data previously on the snapshot repository virtual disk is deleted.
- Snapshot virtual disk and snapshot repository virtual disk parameters remain the same as the previously disabled snapshot virtual disk and snapshot repository virtual disk. You can also change the `userLabel`, `warningThresholdPercent`, and `repositoryFullPolicy` parameters when you restart the snapshot virtual disk.
- The original names for the snapshot repository virtual disk are retained.

Preparing Host Servers To Re-create A Snapshot Virtual Disk

CAUTION: Before you create a new point-in-time image of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk and snapshot virtual disk to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the Snapshot.

Before re-creating a snapshot virtual disk, both the server and the associated virtual disk you are re-creating have to be in the proper state. To ensure that the host server is properly prepared to re-create a snapshot virtual disk, you can either use an application to carry out this task, or you can perform the following steps:

1. Stop all I/O activity to the source and snapshot virtual disk (if mounted).
2. Using your Windows system, flush the cache to both the source and the snapshot virtual disk (if mounted). At the host prompt, type `SMrepassist -f <filename-identifier>` and press **<Enter>**. See "SMrepassist Utility" in the *Owner's Manual* for more information.
3. Remove the drive letter(s) of the source and (if mounted) snapshot virtual disk in Windows or unmount the virtual drive(s) in Linux to help guarantee a stable copy of the drive for the Snapshot. If this is not done, the snapshot operation reports that it has completed successfully, but the snapshot data is not updated properly.
4. Follow any additional instructions for your operating system. Failure to follow these additional instructions can create unusable snapshot virtual disks.

   NOTE: If your operating system requires additional instructions, you can find those instructions in your operating system documentation.

After your server has been prepared, see Re-creating The Snapshot Virtual Disk to re-create the snapshot virtual disk.

Re-creating A Snapshot Virtual Disk

After first preparing the host server(s) as specified in the preceding procedure, use the following examples to re-create a virtual disk snapshot.

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Re-create A Snapshot Virtual Disk. The following example shows the command to restart a snapshot virtual disk:

```
client>smcli 123.45.67.89 -c "recreate snapshot
virtualDisks [\"Mars_Spirit_4-2\" \"Mars_Spirit_4-3\"];"
```

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Re-create A Snapshot Virtual Disk. The following example is the script file version of the command:

```
recreate snapshot virtualDisks ["Mars_Spirit_4-2"
"Mars_Spirit_4-3"];
```

If you do not intend to use a snapshot virtual disk again, you can delete the snapshot virtual disk using the `delete virtualDisk` command. When you delete a snapshot virtual disk, the associated snapshot repository virtual disk is also deleted.

7

Using The Virtual Disk Copy Feature

This chapter describes how the Virtual Disk Copy feature works, lists the script commands for Virtual Disk Copy, and explains how to use the commands to create and run Virtual Disk Copy. Additional information about Virtual Disk Copy and related definitions is available in the online help, the *Deployment Guide*, the *MD Storage Manager online help*, and the *Administrator's Guide*.

NOTE: If you ordered Premium Features for Virtual Disk Copy, you received a Premium Features Activation card shipped in the same box as your Dell PowerVault MD storage array. Follow the directions on the card to obtain a key file and to enable the feature. For more information, see "Premium Feature — Virtual Disk Copy" in the *Owner's Manual*.

The Virtual Disk Copy feature enables you to copy data from one virtual disk (the source) to another virtual disk (the target) in a single storage array. You can use this feature to perform the following functions:

- Back up data.
- Copy data from disk groups that use smaller capacity physical disks to disk groups using larger capacity physical disks.
- Restore snapshot virtual disk data to the associated source virtual disk.
- Copy data from a thin virtual disk to a standard virtual disk on the same storage array.

NOTE: You cannot copy data from a standard virtual disk to a thin virtual disk.

About Virtual Disk Copy

Starting a virtual disk copy operation does the following to your target copy disks:

- Overwrites all existing data on the target virtual disk.
- Makes the target virtual disk read-only to hosts.
- Fails all snapshot (legacy) virtual disks or snapshot image virtual disks associated with the target virtual disk.

If you have data stored on a virtual disk you specify as a virtual disk copy target, make sure you no longer need the data or have it backed up before beginning virtual disk copy.

Virtual Disk Copy Types

The Virtual Disk Copy script commands create one of following types of virtual disk copies:

- A virtual disk copy using a snapshot (legacy), which suspends I/O to the source virtual disk while the copy is in progress. The source virtual disk will not be available during the copy operation. This is called an offline virtual disk copy.
- A virtual disk copy using a point-in-time copy of any virtual disk, while still allowing access to the source virtual disk when the copy is in progress. This is called an online virtual disk copy.

In either type of virtual disk copy, the target virtual disk is locked and cannot be accessed while the copy operation is in place.

After completion of the virtual disk copy of a snapshot (legacy), the legacy snapshot is disabled. After completion of the virtual disk copy using a snapshot image, the snapshot image is deleted and the snapshot virtual disk is disabled.

NOTE: You can have a maximum of eight virtual disk copies in progress at one time. If you try to create more than eight virtual disk copies at one time, the RAID controller modules return a status of Pending until one of the virtual disk copies that is in progress finishes and returns a status of Complete.

NOTE: Snapshots created using older (legacy) premium feature versions cannot be managed using newer snapshot premium feature options. Also, a virtual disk in a snapshot group cannot be a target for a virtual disk copy. If you want to choose the base virtual disk of an older (legacy) snapshot virtual disk as your target virtual disk, you must first disable all snapshot (legacy) virtual disks that are associated with the base virtual disk.

The following table lists the Virtual Disk Copy commands.

NOTE: These commands apply when you are using a snapshot or a snapshot (legacy) image.

Table 12. Virtual Disk Copy Commands

Command	Description
`create virtualDiskCopy`	Creates a virtual disk copy and starts the virtual disk copy operation.
`recopy virtualDiskCopy`	Reinitiates a virtual disk copy operation using an existing virtual disk copy pair.
`enable storageArray feature`	Activates the Virtual Disk Copy feature.
`recopy virtualDiskCopy`	Re-initiates a virtual disk copy operation by using an existing virtual disk copy pair.
`remove virtualDiskCopy`	Removes a virtual disk copy pair.
`set virtualDiskCopy`	Defines the properties for a virtual disk copy pair.
`show virtualDiskCopy`	Returns information about virtual disk copy operations. You can retrieve information about a specific virtual disk copy pair, or all virtual disk copy pairs in the storage array.
`show virtualDiskCopy sourceCandidates`	Returns information about the candidate virtual disks that you can use as the source for a virtual disk copy operation.
`show virtualDiskCopy targetCandidates`	Returns information about the candidate virtual disks that you can use as the target for a virtual disk copy operation.
`stop virtualDiskCopy`	Stops a virtual disk copy operation.

Creating A Virtual Disk Copy

Before creating a virtual disk copy, ensure that a suitable target virtual disk exists on the storage array, or create a new target virtual disk specifically for the virtual disk copy. The target virtual disk must have a capacity equal to or greater than the source virtual disk.

You can have a maximum of eight virtual disk copies in progress at one time. Any virtual disk copy greater than eight has a status of Pending until one of the virtual disk copies with a status of In Progress completes.

The following steps show the general process for creating a virtual disk copy:

1. Enable the Virtual Disk Copy feature.
2. Determine candidates for a virtual disk copy.
3. Create the target virtual disk and source virtual disk for a virtual disk copy.

Enabling The Virtual Disk Copy Feature

The first step in creating a virtual disk copy is to make sure the feature is enabled on the storage array. You need a feature key to enable the feature. To enable the feature key file, use the command:

`enable storageArray feature file="`*`filename`*`"`

where, the `file` parameter is the complete file path and file name of a valid feature key file. Enclose the file path and file name in quotation marks (" "). Valid file names for feature key files usually end with a **.key** extension.

Determining Virtual Disk Copy Candidates

All virtual disks might not be available for use in virtual disk copy operations. To determine which candidate virtual disks on the storage array can be used as a source virtual disk, use the `show virtualDiskCopy sourceCandidates` command. To determine which candidate virtual disks on the storage array can be used as a target virtual disk, use the `show virtualDiskCopy targetCandidates` command. These commands return a list of the expansion enclosure, slot, and capacity information for source virtual disk and target virtual disk candidates. You can use the `show virtualDiskCopy sourceCandidates` and the `show virtualDiskCopy targetCandidates` commands only after you have enabled the virtual disk copy feature.

A source virtual disk can be a standard or thin virtual disk. A target virtual disk can be a standard or thin virtual disk in a disk group or disk pool and, if the legacy version is enabled, a legacy snapshot base virtual disk.

Creating A Virtual Disk Copy

CAUTION: A virtual disk copy overwrites data on the target virtual disk. Ensure that you no longer need the data or have backed up the data on the target virtual disk before starting a virtual disk copy.

When you create a virtual disk copy, you must define which virtual disks to use for the source virtual disk and target virtual disks. Define the source virtual disk and target virtual disk by the name of each virtual disk. You can also define the copy priority and choose whether you want the target virtual disk to be write enabled or read only after the data is copied from the source virtual disk.

Preparing Host Servers To Create A Virtual Disk Copy

CAUTION: Before you create a new copy of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk (and, if applicable, the target disk) to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk(s) in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the virtual disk copy.

Before creating a virtual disk copy, both the server and the associated virtual disk you are copying have to be in the proper state. To ensure that the host server is properly prepared to create a virtual disk copy, you can either use an application to carry out this task, or you can perform the following steps:

1. Stop all I/O activity to the source and target virtual disk.
2. Using your Windows system, flush the cache to both the source and the target virtual disk (if mounted). At the host prompt, type `SMrepassist -f <filename-identifier>` and press **<Enter>**.

   See "SMrepassist Utility" in the Owner's Manual for more information.
3. Remove the drive letter(s) of the source and (if mounted) virtual disk in Windows or unmount the virtual drive(s) in Linux to help guarantee a stable copy of the drive for the virtual disk. If this is not done, the copy operation reports that it has completed successfully, but the copied data is not updated properly.
4. Follow any additional instructions for your operating system. Failure to follow these additional instructions can create unusable virtual disk copies.

   NOTE: If your operating system requires additional instructions, you can find those instructions in your operating system documentation.

After your server has been prepared, see Copying The Virtual Disk to copy the virtual disk.

Copying The Virtual Disk

After first preparing the host server(s) as specified in the preceding procedure, use the following examples to make a virtual disk copy.

The following syntax is the general form of the command:

```
create virtualDiskCopy source="sourceName" target=
"targetName" [copyPriority=(highest | high |
medium | low | lowest) targetReadOnlyEnabled=(TRUE | FALSE)]
```

NOTE: Use one or both of the optional parameters as needed to help define your configuration. It is not necessary to use any optional parameters.

Once the virtual disk copy has started, the source virtual disk is read only to all I/O activity. Any write attempts to the source virtual disk fails until the operation completes.

Once the virtual disk copy operation is completed register the target virtual disk with the operating system to be used by performing the following steps:

- Enable write permission on the target virtual disk by either removing the Virtual Disk Copy Pair or explicitly setting write permission.
  - In Windows, assign a drive letter to the virtual disk.

– In Linux, mount the virtual disk.

See step 1 to step 4 in Preparing Host Servers To Create A Virtual Disk Copy.

The `create virtualDiskCopy` command might look like the following example:

```
client>smcli 123.45.67.89 -c "create
virtualDiskcopy source=\"Jaba_Hut\" target=
\"Obi_1\" copyPriority=medium
targetreadonlyenabled=true"
```

The command in this example copies the data from the source virtual disk named **Jaba_Hut** to the target virtual disk named **Obi_1**. Setting the copy priority to medium provides a compromise between the following storage array operations:

- The speed with which the data is copied from the source virtual disk to the target virtual disk
- The amount of processing resource required for data transfers to other virtual disks in the storage array

Setting the `targetReadOnlyEnabled` parameter to `TRUE` means that write requests cannot be made to the target virtual disk. This setting also ensures that the data on the target virtual disk remains unaltered.

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Create A Virtual Disk Copy. The following example is the script file version of the command:

```
create virtualDiskcopy source="Jaba_Hut" target=
"Obi_1" copyPriority=medium targetreadonlyenabled=true;
```

After the virtual disk copy operation is completed, the target virtual disk automatically becomes read-only to the hosts. Any write requests to the target virtual disk are rejected, unless you disable the read-only attribute. Use the `set virtualDiskCopy` command to disable the read-only attribute.

Viewing Virtual Disk Copy Properties

Using the `show virtualDiskCopy` command, you can view information about one or more selected source virtual disks or target virtual disks. This command returns the following information:

- The virtual disk role (target or source)
- The copy status
- The start timestamp
- The completion timestamp
- The virtual disk copy priority
- The read-only attribute setting for the target virtual disk
- The source virtual disk World Wide Identifier (WWID) or the target virtual disk WWID

A virtual disk can be a source virtual disk for one virtual disk copy and a target virtual disk for another virtual disk copy. If a virtual disk participates in more than one virtual disk copy, the details are repeated for each associated copy pair.

The following syntax is the general form of the command:

```
show virtualDiskCopy (allVirtualDisks | source
[sourceName] | target [targetName])
```

The following example shows a command that returns information about a virtual disk used for a virtual disk copy:

```
client>smcli 123.45.67.89 -c "show virtualDiskCopy
source [\"Jaba_Hut\"];"
```

The command in the preceding example requests information about the source virtual disk **Jaba_Hut**. If you want information about all virtual disks, use the `allVirtualDisks` parameter. You can also request information about a specific target virtual disk.

The following example is the script file version of the command:

```
show virtualDiskCopy source ["Jaba_Hut"];
```

Changing Virtual Disk Copy Settings

The `set virtualDiskCopy` command enables you to change the property settings for a virtual disk copy pair. Using this command, you can change the following items:

- Copy priority
- Read/write permission for the target virtual disk

Copy priority has five relative settings, which range from highest to lowest. The highest priority supports the virtual disk copy, but I/O activity might be affected. The lowest priority supports I/O activity, but the virtual disk copy takes longer. You can change the copy priority at three different times in the operation:

- Before the virtual disk copy begins
- While the virtual disk copy has a status of In Progress
- After the virtual disk copy has completed re-creating a virtual disk copy using the `recopy virtualDiskCopy` command

When you create a virtual disk copy pair and after the original virtual disk copy has completed, the target virtual disk is automatically defined as read-only to the hosts. The read-only status of the target virtual disk ensures that the copied data on the target virtual disk is not corrupted by additional writes to the target virtual disk after the virtual disk copy is created. Maintain the read-only status when the following conditions apply:

- You are using the target virtual disk for backup purposes
- You are copying data from one disk group to a larger disk group for greater accessibility
- You are planning to use the data on the target virtual disk to copy back to the source virtual disk in case of a disabled or failed snapshot virtual disk

At other times you might want to write additional data to the target virtual disk. You can use the `set virtualDiskCopy` command to reset the read/write permission for the target virtual disk.

NOTE: If you enabled host writes to the target virtual disk, read and write requests are rejected while the virtual disk copy has a status of In Progress, Pending, or Failed.

The following syntax is the general form of the command:

```
set virtualDiskCopy target [targetName] [source
[sourceName]] copyPriority=(highest | high |
medium | low | lowest) targetReadOnlyEnabled=(TRUE | FALSE)
```

NOTE: Use one or both of the parameters as needed to help define your configuration. It is not necessary to use either parameter.

The following example shows how to change parameters using the `set virtualDiskCopy` command:

```
client>smcli 123.45.67.89 -c "set virtualDiskcopy
target [\"Obi_1\"] copyPriority=highest
targetreadonlyenabled=false;"
```

The following example is the script file version of the command:

```
set virtualDiskcopy target ["Obi_1"] copyPriority=
highest targetreadonlyenabled=false;
```

Recopying A Virtual Disk

CAUTION: The `recopy virtualDiskCopy` command overwrites existing data on the target virtual disk and makes the target virtual disk read-only to hosts. The `recopy virtualDiskCopy` command fails all snapshot virtual disks associated with the target virtual disk, if any exist.

Using the `recopy virtualDiskCopy` command, you can create a new virtual disk copy for a previously defined copy pair that has a status of Stopped, Failed, or Completed. Use the `recopy virtualDiskCopy` command to create backups of the target virtual disk, then copy the backup to tape for off-site storage. When using the `recopy virtualDiskCopy` command to make a backup, you cannot write to source while the recopy is running. The recopy might take a long time.

When you run the `recopy virtualDiskCopy` command, the data on the source virtual disk is copied in its entirety to the target virtual disk.

Reset the copy priority for the recopy operation by using the `recopy virtualDiskCopy` command. The higher priorities allocate storage array resources to the virtual disk copy at the expense of storage array performance.

Preparing Host Servers To Recopy A Virtual Disk

CAUTION: Before you create a new copy of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk (and, if applicable, the target disk) to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk(s) in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the virtual disk copy.

Before creating a new virtual disk copy for an existing copy pair, both the server and the associated virtual disk you are recopying have to be in the proper state. To ensure that the host server is properly prepared to create a virtual disk recopy, you can either use an application to carry out this task, or you can perform the following steps:

1. Stop all I/O activity to the source and target virtual disk.
2. Using your Windows system, flush the cache to both the source and the target virtual disk (if mounted). At the host prompt, type `SMrepassist -f <filename-identifier>` and press **<Enter>**.

   See "SMrepassist Utility" in the *Owner's Manual* for more information.
3. Remove the drive letter(s) of the source and (if mounted) virtual disk in Windows or unmount the virtual drive(s) in Linux to help guarantee a stable copy of the drive for the virtual disk. If this is not done, the copy operation reports that it has completed successfully, but the copied data is not updated properly.
4. Follow any additional instructions for your operating system. Failure to follow these additional instructions can create unusable virtual disk copies.

NOTE: If your operating system requires additional instructions, you can find those instructions in your operating system documentation.

After your server has been prepared, see Recopying The Virtual Disk to recopy the virtual disk.

Recopying The Virtual Disk

After first preparing the host server(s) as specified in the preceding procedure, use the following examples to make a virtual disk copy.

The following syntax is the general form of the command:

```
recopy virtualDiskCopy target [targetName] [source
[sourceName] copyPriority=(highest | high | medium
| low | lowest) targetReadOnlyEnabled=(TRUE | FALSE)]
```

NOTE: Use one or all of the optional parameters as needed to help define your configuration. It is not necessary to use any optional parameters.

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Recopy A Virtual Disk. The following example shows a command that changes the copy priority:

```
client>smcli 123.45.67.89 -c "recopy
virtualDiskCopy target [\"Obi_1\"] copyPriority=highest;"
```

The command in this example copies data from the source virtual disk associated with the target virtual disk **Obi_1** to the target virtual disk again. The copy priority is set to the highest value to complete the virtual disk copy as quickly as possible. The underlying consideration for using this command is that you have already created the virtual disk copy pair. When you create a virtual disk copy pair, you automatically created one virtual disk copy. Using this command, you are copying the data from the source virtual disk to the target virtual disk. You are making this copy because the data on the source virtual disk changed since the previous copy was made.

Refer to steps 1 through 4 in the preceding section, Preparing Host Servers To Recopy A Virtual Disk. The following example is the script file version of the command:

```
recopy virtualDiskCopy target ["Obi_1"]
copyPriority=highest;
```

Stopping A Virtual Disk Copy

The `stop virtualDiskCopy` command enables you to stop a virtual disk copy that has a status of In Progress, Pending, or Failed. After you stop a virtual disk copy, you can use the `recopy virtualDiskCopy` command to create a new virtual disk copy using the original virtual disk copy pair. All mapped hosts have write access to the source virtual disk.

The following syntax is the general form of the command:

```
stop virtualDiskCopy target [targetName] [source [sourceName]]
```

The following example shows a command to stop a virtual disk copy operation:

```
client>smcli 123.45.67.89 -c "stop virtualDiskCopy target [\"Obi_1\"];"
```

The following example is the script file version of the command:

```
stop virtualDiskCopy target ["Obi_1"];
```

Removing Copy Pairs

The `remove virtualDiskCopy` command enables you to remove a virtual disk copy pair from the storage array configuration. All virtual disk copy information for the source virtual disk and target virtual disk is removed from the storage array configuration. The data on the source virtual disk or target virtual disk is not deleted. Removing a virtual disk copy from the storage array configuration also removes the read-only attribute for the target virtual disk.

CAUTION: If the virtual disk copy has a status of In Progress, you must stop the virtual disk copy before you can remove the virtual disk copy pair from the storage array configuration.

The following syntax is the general form of the command:

```
remove virtualDiskCopy target [targetName] [source [sourceName]]
```

The following example shows a command to remove a virtual disk copy pair:

```
client>smcli 123.45.67.89 -c "remove
virtualDiskCopy target [\"Obi_1\"];"
```

The following example is the script file version of the command:

```
remove virtualDiskCopy target ["Obi_1"];
```

Interaction With Other Features

You can run the Virtual Disk Copy feature while running the following features:

- Storage Partitioning
- Snapshot Virtual Disks

When running the Virtual Disk Copy feature with other features, you must take the requirements of other features into consideration to ensure you set up a stable storage array configuration. You can also run the Virtual Disk Copy feature while running Dynamic Virtual Disk Expansion.

About Snapshot (Legacy) Premium Features With Virtual Disk Copy

- A source virtual disk for a virtual disk copy can be a standard virtual disk or a thin virtual disk (if supported in your firmware version).
- A target virtual disk can be a standard virtual disk in a disk group or disk pool, or a snapshot (legacy) base virtual disk if the snapshot (legacy) premium feature is the only snapshot premium feature enabled on your storage array. A virtual disk in a snapshot group cannot be a virtual disk copy target.
- Snapshots created using the older (legacy) premium feature versions cannot be managed using newer snapshot premium feature options.

8

Using The Remote Replication Premium Feature

The following types of Remote Replication premium features are supported on the MD storage array:

Remote Replication	Standard asynchronous replication using point-in-time images to batch the resynchronization between the local and remote site. This type of replication is supported on either Fibre Channel or iSCSI storage arrays (both local and remote arrays must have the same data protocol).
Remote Replication (Legacy)	Synchronous (or full-write) replication that synchronizes local and remote site data in real-time. This type of replication is supported on Fibre Channel storage arrays only.

NOTE: This chapter describes the standard Remote Replication premium feature supported in both iSCSI and Fibre Channel storage arrays only. To understand the Remote Replication (legacy) premium feature, see Using The Remote Replication (Legacy) Premium Feature.

The Remote Replication premium feature provides for online, real-time replication of data between storage arrays over a remote distance. In the event of a disaster or a catastrophic failure on one storage array, you can promote the second storage array to take over responsibility for computing services. Remote Replication is designed for extended storage environments in which the storage arrays that are used for Remote Replication are maintained at separate sites.

You can use Remote Replication for these functions:

Disaster recovery	Remote Replication lets you replicate data from one site to another site, which provides an exact duplicate at the remote (secondary) site. If the primary site fails, you can use replicated data at the remote site for failover and recovery. You can then shift storage operations to the remote site for continued operation of all of the services that are usually provided by the primary site.
Data vaulting and data availability	Remote Replication lets you send data off site where it can be protected. You can then use the off-site copy for testing or to act as a source for a full backup to avoid interrupting operations at the primary site.
Two-way data protection	Remote Replication provides the ability to have two storage arrays back up each other by duplicating critical virtual disks on each storage array to virtual disks on the other storage array. This action lets each storage array recover data from the other storage array in the event of any service interruptions.

How Remote Replication Works

Standard Remote Replication (asynchronous) is a premium feature that provides RAID controller-based data replication between a local and remote storage array on a per-virtual disk basis. By identifying

primary (local) and secondary (remote) virtual disk pairs, called replicated pairs, write operations to the primary virtual disk of the pair are tracked by the RAID controller firmware and captured in a point-in-time image and transferred to the secondary virtual disk in the pair.

Remote Replication groups allow you to manage synchronization of both virtual disks to create a consistent data set across local and remote storage arrays. Point-in-time images on the primary virtual disk and the secondary virtual disk can be resynchronized in a batch approach that increases replication throughput. When data synchronization completes, the system uses the point-in-time images on the secondary virtual disk to ensure that the data is maintained in a consistent state during subsequent synchronization operations to the secondary virtual disk.

Replication Pairs And Replication Repositories

Replicated pairs, comprising of a primary and secondary virtual disk, contain identical data copies as a result of data synchronization. Replication repository virtual disks are used to manage replication data synchronization and are required for both the primary virtual disk and secondary virtual disk in a replicated pair.

A replication repository consists of the following data:

- Resynchronization and recovery point images for primary and secondary virtual disks.
- Tracking data from the primary virtual disk that occurs in between synchronization intervals. This data is also written to the secondary virtual disk in case a reversal of the primary/secondary virtual disk role should occur.

Differences Between Remote Replication And Remote Replication (Legacy) Features

The standard Remote Replication premium feature uses a point-in-time snapshot image to capture the state of the source virtual disk and only writes data that has changed since the last point-in-time image. The Remote Replication (Legacy) premium feature reproduces every data write to the local (primary) virtual disk on the remote (secondary) virtual disk through a Fibre Channel-only connected configuration. While not producing a fully synchronized data view between both storage arrays, standard Remote Replication offers a much faster replication solution that can run on both iSCSI and Fibre Channel configurations.



NOTE: Both remote and local storage arrays must of the same data protocol. Replication between Fibre Channel and iSCSI storage arrays is not supported.

Standard Remote Replication creates a repository virtual disk for each replicated pair, increasing throughput between local and remote arrays. Remote Replication (Legacy) uses a single repository virtual disk for all replicated pairs.

Physical distance between local and remote storage arrays is unlimited using the Standard Remote Replication premium feature. Remote Replication (legacy) is limited to approximately 10 Km (6.2 miles) between local and remote storage arrays.

Link Interruptions Or Secondary Virtual Disk Errors

When processing write requests, the primary RAID controller module might be able to write to the primary virtual disk, but a link interruption might prevent communication with the remote (secondary) RAID controller module.

In this case, the remote write operation cannot be completed to the secondary virtual disk, and the primary virtual disk and the secondary virtual disk are no longer correctly replicated. The primary RAID controller module transitions the replicated pair into an Unsynchronized state and sends an I/O completion to the primary host. The primary host can continue to write to the primary virtual disk, but remote writes do not take place.

When communication is restored between the RAID controller module owner of the primary virtual disk and the RAID controller module owner of the secondary virtual disk, a resynchronization takes place. This resynchronization happens automatically, or it must be started manually, depending on which write mode you chose when setting up the replication relationship. During the resynchronization, only the blocks of data that have changed on the primary virtual disk during the link interruption are copied to the secondary virtual disk. After the resynchronization starts, the replicated pair transitions from an Unsynchronized status to a Synchronization in Progress status.

The primary RAID controller module also marks the replicated pair as unsynchronized when a virtual disk error on the secondary side prevents the remote write from completing. For example, an offline secondary virtual disk or a failed secondary virtual disk can cause the remote replication to become unsynchronized. When the virtual disk error is corrected (the secondary virtual disk is placed online or recovered to an Optimal status), then synchronization is required. The replicated pair then transitions to a Synchronization in Progress status.

Resynchronization

Data replication between the primary virtual disk and the secondary virtual disk in a replication relationship is managed by the RAID controller modules and is transparent to host machines and applications. When the RAID controller module owner of the primary virtual disk receives a write request from a host, the RAID controller module first logs information about the write to a replication repository virtual disk. The RAID controller module then writes the data to the primary virtual disk. The RAID controller module then initiates a write operation to copy the affected data to the secondary virtual disk on the remote storage array.

If a link interruption or a virtual disk error prevents communication with the secondary storage array, the RAID controller module owner of the primary virtual disk transitions the replicated pair into an Unsynchronized status. The RAID controller module owner then sends an I/O completion to the host sending the write request. The host can continue to issue write requests to the primary virtual disk, but remote writes to the secondary virtual disk do not take place.

When connectivity is restored between the RAID controller module owner of the primary virtual disk and the RAID controller module owner of the secondary virtual disk, the virtual disks must be resynchronized by copying the blocks of data that changed during the interruption to the secondary virtual disk. Only the blocks of data that have changed on the primary virtual disk during the link interruption are copied to the secondary virtual disk.

△ **CAUTION: Possible loss of data access – Any communication disruptions between the primary storage array and the secondary storage array while resynchronization is underway could result in a mix of new data and old data on the secondary virtual disk. This condition would render the data unusable in a disaster recovery situation.**

Remote Replication Group

After your Remote Replication premium feature is activated on both local and remote storage arrays, you must create Remote Replication groups on the local storage array.

A replication group contains at least one replicated virtual disk pair, one on the local storage and one on the remote storage array, that share data synchronization settings. Multiple replicated pairs can reside in a replication group, but each pair can only be a member of one Remote Replication group.

The following attributes also apply to a Remote Replication group:

- The local storage array serves as the primary side of the Remote Replication group, while the remote storage array serves as the secondary side of the Remote Replication group.
- At the virtual disk level, all virtual disks added to the Remote Replication group on the local storage array serve as the primary role in the Remote Replication configuration. Virtual disks added to the group on the remote storage array serve the secondary role.

Once you have selected virtual disks at both the remote and local storage array that you want to pair in a replication relationship, adding them to a replication group actually begins the replication synchronization process.

For more detailed information on the role of Remote Replication groups, see the *Administrator's Guide*.

Previous Users Of Remote Replication (Legacy) Premium Feature

If you have upgraded (or plan to upgrade) your RAID controller firmware version to a level that supports both legacy and non-legacy Remote Replication, any legacy replication configurations you have previously set up will be unaffected, and will continue to function normally.

Remote Replication Requirements And Restrictions

To use the standard Remote Replication premium feature, you must have:

- Two storage arrays with write access and both these storage arrays must have sufficient space to replicate data between them.
- Each storage must have a dual-controller Fibre Channel or iSCSI configuration (single-controller configurations are not supported).
- Fibre Channel Connection Requirements — You must attach dedicated remote replication ports to a Fibre Channel fabric environment. In addition, these ports must support the Name Service.
- You can use a fabric configuration that is dedicated solely to the remote replication ports on each RAID controller module. In this case, host systems can connect to the storage arrays using fabric.
- Fibre Channel Arbitrated Loop (FC-AL), or point-to-point configurations, are not supported for array-to-array communications.
- Maximum distance between the local site and remote site is 10 km (6.2 miles), using single-mode fibre Gigabit interface converters (GBICs) and optical long-wave GBICs.
- iSCSI connection considerations:
  - iSCSI does not require dedicated ports for replication data traffic
  - iSCSI array-to-array communication must use a host-connected port (not the Ethernet management port).

– The first port that successfully establishes an iSCSI connection is used for all subsequent communication with that remote storage array. If that connection subsequently fails, a new session is attempted using any available ports.

Primary And Secondary Virtual Disks

Before you create any replication relationships, virtual disks must exist at both the primary site and the secondary site. The virtual disk that resides on the local storage array is the primary virtual disk. Similarly, the virtual disk that resides on the remote storage array is the secondary virtual disk. If neither the primary virtual disk nor the secondary virtual disk exist, you must create these virtual disks. Keep these guidelines in mind when you create the secondary virtual disk:

- The secondary virtual disk must be of equal or greater size than the primary virtual disk.
- The RAID level of the secondary virtual disk does not have to be the same as the primary virtual disk.

Setting Up Remote Replication

Setting up Remote Replication between local and remote storage arrays consists of the following:

- Enabling Remote Replication (on both storage arrays)
- Activating the Remote Replication premium feature on both the local and remote storage arrays.
- Creating a Remote Replication group on the local storage array.
- Adding a replicated pair of virtual disks to the Remote Replication group.

Enabling The Remote Replication Premium Feature

The first step in creating a remote replication is to make sure that the Remote Replication premium feature is enabled on both storage arrays. Because Remote Replication is a premium feature, you need a feature key file to enable the premium feature. The command for enabling the feature key file is as follows:

```
enable storageArray feature file=”asyncReplicationactivation_key”
```

In this command, `asyncReplication` is literal and is appended with the *activation_key*, which is provided by Dell. For example, if your activation key value is 999999:

```
enable storageArray feature file="asyncReplication999999"
```

Activating The Remote Replication Premium Feature

Activating the Remote Replication premium feature prepares the storage arrays to create and configure replication relationships. After you activate the premium feature, the secondary ports for each RAID controller module are reserved and dedicated to remote replication use. In addition, replication repository virtual disks are automatically created for each RAID controller module in the storage array.

To activate the Remote Replication premium feature, use this command:

```
activate storageArray feature=asyncReplication
```

Creating A Remote Replication Group

The first step in establishing a Remote Replication relationship is to create a replication group on the local storage array. A replication group cannot be created on the remote storage array.

The `create asyncRemoteReplicationGroup` command creates a new, empty remote replication group on both the local storage array and the remote storage array. Each replicated pair you add to the remote replication group share the same synchronization settings, primary and secondary role, and write mode.

This command must be run on the local storage array. Remote replication group creation is initiated from the storage array that contains the virtual disks that hold the primary role in the replicate relationship. The following command create a replication group names **RRG-001** on a remote iSCSI storage array named **Remote_SS_A101**. The remote storage array has a password of **123Dell321** (remotePassword=). A warning will be triggered when the capacity of a replication repository virtual disk reaches 75 percent of capacity (warningThresholdPercent=).

```
create asyncRemoteReplicationGroup userLabel="RRG-001"
remoteStorageArrayName="Remote_SS_A101" interfaceType=iSCSI
remotePassword="123Dell321"  warningThresholdPercent=75;
```

Adding Primary Virtual Disk To Remote Replication Group

The `add virtualDisk` command adds a primary virtual disk to a remote replication group. This command is valid only on the local storage array that contains the remote replication group to which you want to add the primary virtual disk. To add the secondary virtual disk on the remote storage array to the remote replication group, use the `establish asyncRemoteReplication virtualDisk` command.

You have two options for specifying a repository virtual disk when using this command: you can either specify an existing repository virtual disk or create a new one when running the command.

To add a virtual disk named **employeeBackfilData** to the **Remote_SS_A101** group you created using an existing repository virtual disk named **rep_VD_404**:

```
add virtualDisk="employeeBackfilData"
asyncRemoteReplicationGroup="Remote_SS_A101"
repositoryVirtualDisk="rep_VD_404";
```

To add the same virtual disk to the same replication, but create a new 10GB repository virtual disk from an existing disk group named **DG_09**:

```
add virtualDisk="employeeBackfilData"
asyncRemoteReplicationGroup="Remote_SS_A101"
repositoryVirtualDisk=DG_09 capacity=10GB;
```

You can substitute a disk pool name for the disk group name, if creating the repository virtual disk from disk pool space.

Changing Remote Replication Group Settings

The `set asyncRemoteReplicationGroup` command lets you change settings for an existing replication group.

The following parameters set synchronization and warning threshold values for the replication group. Changing the synchronization settings affects the synchronization operations of all replicated pairs within the remote replication group.

- `syncInterval`
- `warningSyncThreshold`
- `warningRecoveryThreshold`

- `warningThresholdPercent`

The following parameters allow you to change (or force) the primary/secondary role of a replication group, or whether or not to perform a synchronization before changing primary/secondary roles.

- `role`
- `force`
- `nosync`

You can apply the changes to one or several remote replicated pairs by using this command. Use the primary virtual disk name to identify the remote replicated pairs for which you are changing the properties.

The full command syntax is shown below:

```
set asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
[syncInterval=integer (minutes | hours | days)
warningSyncThreshold=integer (minutes | hours | days)
warningRecoveryThreshold=integer (minutes | hours | days)
warningThresholdPercent=percentValue  autoResync=(TRUE | FALSE)
virtualDisk="virtualDiskName"
increaseRepositoryCapacity (repositoryVirtualDisk="repos_xxxx" |
repositoryVirtualDisk=(diskGroupName [capacity=capacityValue])
repositoryVirtualDisk=(diskPoolName [capacity=capacityValue])) role=(primary|
secondary) ([force=TRUE|FALSE]|[noSync=TRUE|FALSE])
```

Adding Secondary Virtual Disk To Remote Replication Group

Use the `establish asyncRemoteReplication` command to add the secondary virtual disk on the remote storage array to the replication group. This command completes the remote replicated pair process begun with the `add virtualDisk` command, which added the primary virtual disk on the local storage array to the replication group.

Before running this command, the remote replication group must exist and the primary virtual disk must exist in the remote replication group. After the `establish asyncRemoteReplication` command successfully completes, remote replication automatically starts between the primary virtual disk and the secondary virtual disk.

Using the previous example, the following command completes a replicated pair within a replication group named **Remote_SS_A101** between the primary virtual disk (on the local storage array) named **employeeBackfilData** and a secondary virtual disk (on the remote storage array) named **employeeBackfilData_remote**:

```
establish asyncRemoteReplication
virtualDisk="employeeBackFilData_remote"
asyncRemoteReplicationGroup="Remote_SS_A101"
primaryVirtualDisk="employeeBackfilData";
```

Suspending A Remote Replication Group

Use the `asyncRemoteReplicationGroup` command to stop data transfer between a primary virtual disk and a secondary virtual disk in a replication relationship without disabling the replication relationship. Suspending a replication relationship group lets you control when the data on the primary virtual disk and data on the secondary virtual disk are synchronized. Suspending a replication relationship group helps to

reduce any performance impact to the host application that might occur while any changed data on the primary virtual disk is copied to the secondary virtual disk.

When a replication relationship is in a suspended state, the primary virtual disk does not make any attempt to contact the secondary virtual disk. Any writes to the primary virtual disk are persistently logged in the replication repository virtual disks. After the replication relationship resumes, any data that is written to the primary virtual disk is automatically written to the secondary virtual disk. Only the modified data blocks on the primary virtual disk are written to the secondary virtual disk. Full synchronization is not required.

This example shows the `suspend remoteReplicationGroup` command:

```
suspend asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"];
```

The replication relationship group remains suspended until you use the `resume asyncRemoteReplicationGroup` command to restart synchronization activities.

This example shows the `resume remoteReplicationGroup` command:

```
resume asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"];
```

Deleting A Remote Replication Group

Use the `delete asyncRemoteReplicationGroup` command to delete one or more replication groups from the local or remote storage array. The replication group you are attempting to delete must be empty (contain no virtual disks or replicated pairs) before running this command.

This example shows the `delete asyncRemoteReplicationGroup` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c delete asyncRemoteReplicationGroup["replication_group_name"];
```

To delete all replication groups from the local and remote storage arrays, use the `delete allAsyncRemoteReplicationGroups` command.

To delete selected replication groups, separate group names with "":

```
delete asyncRemoteReplicationGroups ["name1""name2"];
```

Removing A Virtual Disk Or Repository Virtual Disk From A Remote Replication Group

Use the `remove virtualDisk` command to remove either a member virtual disk or repository virtual disk from an existing replication group. This command can only be run on the local storage array containing the replication group affected by the command.

The command shown below removes *diskname* from the replication group named *groupname*. Using the optional `deleteRepositoryMembers` parameter with a value of `TRUE` also delete the repository virtual disk members:

```
remove virtualDisk ["diskname"]
asyncRemoteReplicationGroup="groupname"
deleteRepositoryMembers=TRUE;
```

Disabling The Remote Replication Premium Feature

You disable the Remote Replication premium feature to prevent the new replication relationship from being created. When you disable the Remote Replication premium feature, the premium feature is in a Disabled/Active state. In this state, you can maintain and manage previously existing replication relationships; however, you cannot create new relationships. To disable the Remote Replication premium feature, use this command:

```
disable storageArray feature=asyncReplication;
```

Deactivating The Remote Replication Premium Feature

If you no longer require the Remote Replication premium feature and you have removed all of the replication relationships (replication groups and replicated pairs), you can deactivate the premium feature. Deactivating the premium feature re-establishes the normal use of dedicated ports on both storage arrays and deletes replication repository virtual disks. To deactivate the Remote Replication premium feature, use this command:

```
deactivate storageArray feature=asyncReplication;
```

Interaction With Other Premium Features

You can run the Remote Replication premium feature while running these premium features:

- Snapshot—both standard Snapshot and Snapshot (legacy) premium features
- Virtual Disk Copy

When you run the Remote Replication (legacy) premium feature with other premium features, you must consider the requirements of the other premium features to ensure that you set up a stable storage array configuration.

For more information on using Remote Replication with other premium features, see the *Administrator's Guide*.

Standard Remote Replication Commands

The following Remote Replication commands are also available. For more information, see Commands Listed Alphabetically.

Table 13. Standard Remote Replication Commands

Command	Description
`stop asyncRemoteReplicationGroup rolechange`	Cancels a pending primary/secondary storage array role reversal.
`check asyncRemoteReplicationGroup repositoryConsistency`	Reports information on data in the repository virtual disks.
`clear asyncRemoteReplicationFault`	Clears an remote replication "sticky" fault from a replication group(s) and/or member virtual disk.
`check storageArray connectivity`	Verifies the local and the remote storage arrays have a valid communication path.

Command	Description
`remove asyncRemoteReplicationGroup incompleteRemoteReplication`	Removes an orphaned replicated pair virtual disk.
`reset storageArray arvmStats reset storageArray iscsiIpAddress`	Resets IP address for the remote storage array to re-establish connection with the local storage array or resets synchronization statistics for member virtual disks to relative 0.
`show asyncRemoteReplicationGroup synchronizationProgress`	Shows progress of periodic synchronization of the replication group as a percentage.
`show asyncRemoteReplicationGroup summary`	Shows configuration information for one or more replication groups.

9

Using The Remote Replication (Legacy) Premium Feature

The Remote Replication (legacy) premium feature provides for online, real-time replication of data between storage arrays over a remote distance. In the event of a disaster or a catastrophic failure on one storage array, you can promote the second storage array to take over responsibility for computing services. Remote Replication (legacy) is designed for extended storage environments in which the storage arrays that are used for Remote Replication (legacy) are maintained at separate sites. Virtual disks on one storage array are replicated to virtual disks on another storage array across a fabric SAN. Data transfers can be synchronous or asynchronous. You choose the method when you set up the remote replicated pair. The data transfers occur at Fibre Channel speeds to maintain data on the different storage arrays. Because Remote Replication (legacy) is storage based, it does not require any server overhead or application overhead.

You can use Remote Replication (legacy) for these functions:

Disaster recovery	Remote Replication (legacy) lets you replicate data from one site to another site, which provides an exact duplicate at the remote (secondary) site. If the primary site fails, you can use replicated data at the remote site for failover and recovery. You can then shift storage operations to the remote site for continued operation of all of the services that are usually provided by the primary site.
Data vaulting and data availability	Remote Replication (legacy) lets you send data off site where it can be protected. You can then use the off-site copy for testing or to act as a source for a full backup to avoid interrupting operations at the primary site.
Two-way data protection	Remote Replication (legacy) provides the ability to have two storage arrays back up each other by duplicating critical virtual disks on each storage array to virtual disks on the other storage array. This action lets each storage array recover data from the other storage array in the event of any service interruptions.

How Remote Replication (Legacy) Works

When you create a remote replicated pair, the remote replicated pair consists of a primary virtual disk on a local storage array and a secondary virtual disk on a storage array at another site. A standard virtual disk might only be included in one replicated virtual disk pair.

The primary virtual disk is the virtual disk that accepts host I/O activity and stores application data. When the replication relationship is first created, data from the primary virtual disk is copied in its entirety to the secondary virtual disk. This process is known as a full synchronization and is directed by the RAID controller module owner of the primary virtual disk. During a full synchronization, the primary virtual disk remains fully accessible for all normal I/O operations.

The RAID controller module owner of the primary virtual disk initiates remote writes to the secondary virtual disk to keep the data on the two virtual disks synchronized.

The secondary virtual disk maintains a replication (or copy) of the data on its associated primary virtual disk. The RAID controller module owner of the secondary virtual disk receives remote writes from the RAID controller module owner of the primary virtual disk but will not accept host write requests. Hosts are able to read from the secondary virtual disk, which appears as read-only.

In the event of a disaster or a catastrophic failure at the primary site, you can perform a role reversal to promote the secondary virtual disk to a primary role. Hosts then are able to read from and write to the newly promoted virtual disk, and business operations can continue.

Replication Repository Virtual Disks

A replication repository virtual disk is a special virtual disk in the storage array that is created as a resource for the RAID controller module owner of the primary virtual disk in a remote replicated pair. The RAID controller module stores replication information on this virtual disk, including information about remote writes that are not yet complete. The RAID controller module can use this information to recover from RAID controller module resets and the accidental powering down of the storage arrays.

When you activate the Remote Replication (legacy) premium feature on the storage array, you create two replication repository virtual disks, one for each RAID controller module in the storage array. An individual replication repository virtual disk is not needed for each remote replication.

When you create the replication repository virtual disks, you specify the location of the virtual disks. You can either use existing free capacity, or you can create a disk group for the virtual disks from unconfigured capacity and then specify the RAID level.

Because of the critical nature of the data being stored, do not use RAID Level 0 as the RAID level of replication repository virtual disks. The required size of each virtual disk is 128 MB, or 256 MB total for both replication repository virtual disks of a dual-RAID controller module storage array. In previous versions of the Remote Replication (legacy) premium feature, the replication repository virtual disks required less disk storage space and needed to be upgraded to use the maximum amount of replication relationships.

Replication Relationships

Before you create a replication relationship, you must enable the Remote Replication (legacy) premium feature on both the primary storage array and the secondary storage array. You must also create a secondary virtual disk on the secondary site if one does not already exist. The secondary virtual disk must be a standard virtual disk of equal or greater capacity than the associated primary virtual disk.

When secondary virtual disks are available, you can establish a replication relationship in the MD storage management software by identifying the primary virtual disk and the storage array that contains the secondary virtual disk.

When you first create the replication relationship, a full synchronization automatically occurs, with data from the primary virtual disk copied in its entirety to the secondary virtual disk.

Data Replication

The RAID controller modules manage data replication between the primary virtual disk and the secondary virtual disk. This process is transparent to host machines and applications. This section describes how data is replicated between the storage arrays that are participating in Remote Replication (legacy). This section also describes the actions taken by the RAID controller module owner of the primary virtual disk if a link interruption occurs between storage arrays.

Write Modes

When the RAID controller module owner of the primary virtual disk receives a write request from a host, the RAID controller module first logs information about the write to a replication repository virtual disk, and then writes the data to the primary virtual disk. The RAID controller module then initiates a remote write operation to copy the affected data blocks to the secondary virtual disk at the secondary storage array.

The Remote Replication (legacy) premium feature provides two write mode options that affect when the I/O completion indication is sent back to the host: Synchronous and Asynchronous.

Synchronous Write Mode

Synchronous write mode provides the highest level security for full data recovery from the secondary storage array in the event of a disaster. Synchronous write mode does reduce host I/O performance. When this write mode is selected, host write requests are written to the primary virtual disk and then copied to the secondary virtual disk. After the host write request has been written to the primary virtual disk and the data has been successfully copied to the secondary virtual disk, the RAID controller module removes the log record on the replication repository virtual disk. The RAID controller module then sends an I/O completion indication back to the host system. Synchronous write mode is selected as the default value and is the recommended write mode.

Asynchronous Write Mode

Asynchronous write mode offers faster host I/O performance but does not guarantee that a copy operation has successfully completed before processing the next write request. When you use Asynchronous write mode, host write requests are written to the primary virtual disk. The RAID controller module then sends an "I/O complete" indication back to the host system, without acknowledging that the data has been successfully copied to the secondary (remote) storage array.

When using Asynchronous write mode, write requests are not guaranteed to be completed in the same order on the secondary virtual disk as they are on the primary virtual disk. If the order of write requests is not retained, data on the secondary virtual disk might become inconsistent with the data on the primary virtual disk. This event could jeopardize any attempt to recover data if a disaster occurs on the primary storage array.

Write Consistency Mode

When multiple replication relationships exist on a single storage array and have been configured to use Asynchronous write mode and to preserve consistent write order, they are considered to be an interdependent group that is in the Write consistency mode. The data on the secondary, remote storage array cannot be considered fully synchronized until all of the remote replications that are in the Write consistency mode are synchronized.

If one replication relationship in the group becomes unsynchronized, all of the replication relationships in the group become unsynchronized. Any write activity to the remote, secondary storage arrays is prevented to protect the consistency of the remote data set.

Link Interruptions Or Secondary Virtual Disk Errors

When processing write requests, the primary RAID controller module might be able to write to the primary virtual disk, but a link interruption might prevent communication with the remote (secondary) RAID controller module.

In this case, the remote write operation cannot be completed to the secondary virtual disk, and the primary virtual disk and the secondary virtual disk are no longer correctly replicated. The primary RAID controller module transitions the replicated pair into an Unsynchronized state and sends an I/O completion to the primary host. The primary host can continue to write to the primary virtual disk, but remote writes do not take place.

When communication is restored between the RAID controller module owner of the primary virtual disk and the RAID controller module owner of the secondary virtual disk, a resynchronization takes place. This resynchronization happens automatically, or it must be started manually, depending on which write mode you chose when setting up the replication relationship. During the resynchronization, only the blocks of data that have changed on the primary virtual disk during the link interruption are copied to the secondary virtual disk. After the resynchronization starts, the replicated pair transitions from an Unsynchronized status to a Synchronization in Progress status.

The primary RAID controller module also marks the replicated pair as unsynchronized when a virtual disk error on the secondary side prevents the remote write from completing. For example, an offline secondary virtual disk or a failed secondary virtual disk can cause the remote replication to become unsynchronized. When the virtual disk error is corrected (the secondary virtual disk is placed online or recovered to an Optimal status), then synchronization is required. The replicated pair then transitions to a Synchronization in Progress status.

Resynchronization

Data replication between the primary virtual disk and the secondary virtual disk in a replication relationship is managed by the RAID controller modules and is transparent to host machines and applications. When the RAID controller module owner of the primary virtual disk receives a write request from a host, the RAID controller module first logs information about the write to a replication repository virtual disk. The RAID controller module then writes the data to the primary virtual disk. The RAID controller module then initiates a write operation to copy the affected data to the secondary virtual disk on the remote storage array.

If a link interruption or a virtual disk error prevents communication with the secondary storage array, the RAID controller module owner of the primary virtual disk transitions the replicated pair into an Unsynchronized status. The RAID controller module owner then sends an I/O completion to the host sending the write request. The host can continue to issue write requests to the primary virtual disk, but remote writes to the secondary virtual disk do not take place.

When connectivity is restored between the RAID controller module owner of the primary virtual disk and the RAID controller module owner of the secondary virtual disk, the virtual disks must be resynchronized by copying the blocks of data that changed during the interruption to the secondary virtual disk. Only the

blocks of data that have changed on the primary virtual disk during the link interruption are copied to the secondary virtual disk.

CAUTION: Possible loss of data access – Any communication disruptions between the primary storage array and the secondary storage array while resynchronization is underway could result in a mix of new data and old data on the secondary virtual disk. This condition would render the data unusable in a disaster recovery situation.

Creating A Remote Replicated Pair

Before you create any replication relationships, virtual disks must exist at both the primary site and the secondary site. The virtual disk that resides on the local storage array is the primary virtual disk. Similarly, the virtual disk that resides on the remote storage array is the secondary virtual disk. If neither the primary virtual disk nor the secondary virtual disk exist, you must create these virtual disks. Keep these guidelines in mind when you create the secondary virtual disk:

- The secondary virtual disk must be of equal or greater size than the primary virtual disk.
- The RAID level of the secondary virtual disk does not have to be the same as the primary virtual disk.

Use these steps to create the virtual disk.

1. Enable the Remote Replication (legacy) premium feature.
2. Activate the Remote Replication (legacy) premium feature.
3. Determine candidates for a remote replicated pair.
4. Create the remote relationship.

Performance Considerations

Keep these performance considerations in mind when you create replication relationships:

- The RAID controller module owner of a primary virtual disk performs a full synchronization in the background while processing local I/O writes to the primary virtual disk and associated remote writes to the secondary virtual disk. Because the full synchronization diverts RAID controller module processing resources from I/O writes, full synchronization can have a performance impact to the host application.
- To reduce the performance impact, you can set the synchronization priority level to determine how the RAID controller module owner will prioritize the full synchronization relative to other I/O activity. To set the synchronization priority level, consider these guidelines:
  - A full synchronization at the lowest synchronization priority level takes approximately eight times as long as a full synchronization at the highest synchronization priority level.
  - A full synchronization at the low synchronization priority level takes approximately six times as long as a full synchronization at the highest synchronization priority level.
  - A full synchronization at the medium synchronization priority level takes approximately three-and-a-half times as long as a full synchronization at the highest synchronization priority level.
  - A full synchronization at the high synchronization priority level takes approximately twice as long as a full synchronization at the highest synchronization priority level.
- When the replicated virtual disk pair is in a Synchronization in Progress state, all host write data is copied to the remote system. Both RAID controller module I/O bandwidth and I/O latency can affect host write performance. Host read performance is not affected by the replication relationship.
- The time that it takes for data to be copied from the primary virtual disk to the secondary virtual disk might impact overall performance. This impact is primarily caused by the delay and system resource required for copying data to the remote replication. Some delay might also occur because of the limit to the number of simultaneous writes.

Enabling The Remote Replication (Legacy) Premium Feature

The first step in creating a remote replication is to make sure that the Remote Replication (legacy) premium feature is enabled on both storage arrays. Because Remote Replication (legacy) is a premium feature, you need a feature key file to enable the premium feature. The command for enabling the feature key file is as follows:

```
enable storageArray feature file="filename"
```

In this command, the `file` parameter is the complete file path and file name of a valid feature key file. Enclose the file path and the file name in double quotation marks (" "). Valid file names for feature key files end with a **.key** extension.

Activating The Remote Replication (Legacy) Premium Feature

Activating the Remote Replication (legacy) premium feature prepares the storage arrays to create and configure replication relationships. After you activate the premium feature, the secondary ports for each RAID controller module are reserved and dedicated to remote replication use. In addition, a replication repository virtual disk is automatically created for each RAID controller module in the storage array. As part of the activation process, you can decide where the replication repository virtual disks will reside, free capacity on an existing disk group or in a newly created disk group, and the RAID level for the replication repository virtual disks.

The free capacity that you select for the replication repository virtual disk must have a total of 256 MB of capacity available. Two replication repository virtual disks are created on this capacity, one for each RAID controller module. If you enter a value for the repository storage space that is too small for the replication repository virtual disks, the firmware returns an error message that gives the amount of space needed for the replication repository virtual disks. The command does not try to activate the Remote Replication (legacy) premium feature. You can re-enter the command using the value from the error message for the repository storage space value.

The RAID level that you choose for the replication repository virtual disk has these constraints:

RAID Level 0	You cannot use RAID Level 0.
RAID Level 1	The number of physical disks must be an even number. If you select an odd number of physical disks, the RAID controller module firmware returns an error.
RAID Level 5	You must have a minimum of three physical disks in the disk group.
RAID Level 6	You must have a minimum of five physical disks in the disk group.

To activate the Remote Replication (legacy) premium feature, use this command:

```
activate storageArray feature=remoteReplication
```

The `activate storageArray feature=remoteReplication` command provides three methods for defining the physical disks for your replication repository virtual disk:

- You define each physical disk for the replication repository virtual disk by its enclosure ID and its slot ID.
- You define a disk group in which the replication repository virtual disk resides. You can optionally define the capacity of the replication repository virtual disk.

- You define the number of physical disks, but not specific physical disks, for the replication repository virtual disk.

Activating The Remote Replication (Legacy) Premium Feature With User-Assigned Physical Disks

Activating the Remote Replication (legacy) premium feature by assigning the physical disks provides flexibility in defining your configuration by letting you choose from the available physical disks in your storage array. Choosing the physical disks for your remote replication automatically creates a new disk group. You can specify which physical disks to use and the RAID level for the new disk group.

The command takes this form:

```
activate storageArray feature=remoteReplication

repositoryRAIDLevel=(1 | 5 | 6)

repositoryPhysicalDisks=(enclosureID1,slotID1 ...
enclosureIDn,slotIDn)

enclosureLossProtect=(TRUE | FALSE)
```

This example shows a command in which you assign the physical disks:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "activate storageArray feature=remoteReplication
repositoryRAIDLevel=5
repositoryPhysicalDisks=(1,1 1,2 1,3 1,4 1,5);"
```

The command in this example creates a new replication repository virtual disk consisting of five physical disks that form a new disk group. The new disk group has RAID Level 5.

This example shows how to use the command in a script file:

```
activate storageArray feature=remoteReplication
repositoryRAIDLevel=5
repositoryPhysicalDisks=(1,1 1,2 1,3 1,4 1,5);
```

Activating The Remote Replication (Legacy) Premium Feature With Software-Assigned Physical Disks

With this version of the `activate storageArray feature= remoteReplication` command, you choose an existing disk group in which to place the replication repository virtual disk. The MD storage management software then determines which physical disks to use. You can also define how much space to assign to the replication repository virtual disk. Because you are using an existing disk group, the RAID level for the replication repository virtual disk defaults to the RAID level of the disk group in which you place it. You cannot define the RAID level for the replication repository virtual disk.

The command takes this form:

```
activate storageArray feature=remoteReplication
repositoryDiskGroup=diskGroupNumber
[freeCapacityArea=freeCapacityIndexNumber]
```

This example shows a command in which the software assigns the physical disks:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "activate storageArray feature=remoteReplication
repositoryDiskGroup=2 freeCapacityArea=2;"
```

The command in this example creates a new replication repository virtual disk in disk group 2 using the second free capacity area.

This example shows how to use the command in a script file:

```
activate storageArray feature=remoteReplication
repositoryDiskGroup=2 freeCapacityArea=2;
```

Activating The Remote Replication (Legacy) Premium Feature By Specifying A Number Of Physical Disks

With this version of the `activate storageArray feature= remoteReplication` command, you must specify the number of physical disks and the RAID level that you want for the replication repository virtual disk. This version of the command creates a new disk group. For this command to work, you must have physical disks in the storage array that are not assigned to a disk group.

```
activate storageArray feature=remoteReplication
repositoryRAIDLevel=(1 | 5 | 6)
repositoryPhysicalDiskCount=numberOfPhysicalDisks
[physicalDiskType=(SAS)]
[enclosureLossProtect=(TRUE | FALSE)]
```

This example shows a command in which you specify the number of physical disks:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "activate storageArray feature=remoteReplication
repositoryRAIDLevel=5 repositoryPhysicalDiskCount=5
physicalDiskType=SAS;"
```

The command in this example creates a new replication repository virtual disk by using five software-selected physical disks for the replication repository virtual disk. The replication repository virtual disk has RAID Level 5. The type of physical disk for the replication repository virtual disk is SAS.

This example shows how to use the command in a script file:

```
activate storageArray feature=remoteReplication
repositoryRAIDLevel=5 repositoryCount=5
physicalDiskType=SAS;
```

Determining Candidates For A Remote Replicated Pair

All of the virtual disks and physical disks on the remote storage array might not be available for use as secondary virtual disks. To determine which virtual disks on a remote storage array that you can use as candidates for secondary virtual disks, use the `show remoteReplication candidates` command. This command returns a list of the virtual disks that you can use when creating a remote replication.

The command takes this form:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "show remoteReplication candidates primary= \"virtualDiskName\"
remoteStorageArrayName=\"storageArrayName\";"
```

where `virtualDiskName` is the name of the virtual disk that you want to use for the primary virtual disk, and `storageArrayName` is the remote storage array that contains possible candidates for the secondary virtual disk. Enclose both the virtual disk name and the storage array name in double quotation marks (" ").

Creating A Remote Replicated Pair

When you create a new remote replication, you must define which virtual disks that you want to use for the primary (local) virtual disk and the secondary (remote) virtual disk. You define the primary virtual disk

by the name of the virtual disk. You define the secondary virtual disk by name with either the name or the World Wide Identifier (WWID) of the storage array on which the secondary virtual disk resides. The primary virtual disk name, the secondary virtual disk name, and the remote storage array name (or WWID) are the minimum information that you need to provide. Using this command, you can also define synchronization priority, write order, and write mode.

The command takes this form:

```
create remoteReplication primary="primaryVirtualDiskName"
secondary="secondaryVirtualDiskName"
(remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwn="wwID") remotePassword=password
syncPriority=(highest | high | medium | low | lowest)
writeOrder=(preserved | notPreserved)
writeMode=(synchronous | asynchronous)
```

NOTE: You can use the optional parameters as needed to help define your configuration.

This example shows the `create remoteReplication` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "create remoteReplication primary=
\"Jan_04_Account\"
secondary=\"Jan_04_Account_B\"
remoteStorageArrayName=\"Tabor\"
remotePassword=\"jdw2ga05\" syncPriority=highest
writeMode=synchronous;"
```

The command in this example creates a remote replication in which the primary virtual disk is named **Jan_04_Account** on the local storage array. The secondary virtual disk is named **Jan_04_Account_B** on the remote storage array that is named **Tabor**. The names used in this example are similar, but that is not a requirement for the virtual disk names in a remote replicated pair. In this example, the remote storage array has a password that you must enter when making any change to the storage array configuration. Creating a remote replicated pair is a significant change to a storage array configuration.

Setting the write mode to synchronous and the synchronization priority to highest means that host write requests are written to the primary virtual disk and then immediately copied to the secondary virtual disk. These actions help to make sure that the data on the secondary virtual disk is as accurate a copy of the data on the primary virtual disk as possible. The highest synchronization priority does, however, use more system resources, which can reduce system performance.

This example shows how to use the command in a script file:

```
create remoteReplication primary="Jan_04_Account"
secondary="Jan_04_Account_B" remoteStorageArrayName="Tabor"
remotePassword="jdw2ga05" syncPriority=highest
writeMode=synchronous;
```

After you have created a remote replication, you can see the progress of data synchronization between the primary virtual disk and the secondary virtual disk by running the `show remoteReplication synchronizationProgress` command. This command shows the progress as a percentage of data synchronization that has completed.

Changing Remote Replication (Legacy) Settings

The `set remoteReplication` command lets you change the property settings for a remote replicated pair. Use this command to change these property settings:

- The virtual disk role (either primary or secondary)

- The synchronization priority
- The write order
- The write mode

You can apply the changes to one or several remote replicated pairs by using this command. Use the primary virtual disk name to identify the remote replicated pairs for which you are changing the properties.

This example shows how to use the `set remoteReplication` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "set remoteReplication localVirtualDisk [Jan_04_Account]
syncPriority=medium
writeOrder=notpreserved
writeMode=asynchronous;"
```

This example shows how to use the command in a script file:

```
set remoteReplication localVirtualDisk [Jan_04_Account]
syncPriority=medium
writeOrder=notpreserved
writeMode=asynchronous;
```

Suspending And Resuming A Replication Relationship

Use the `suspend remoteReplication` command to stop data transfer between a primary virtual disk and a secondary virtual disk in a replication relationship without disabling the replication relationship. Suspending a replication relationship lets you control when the data on the primary virtual disk and data on the secondary virtual disk are synchronized. Suspending a replication relationship helps to reduce any performance impact to the host application that might occur while any changed data on the primary virtual disk is copied to the secondary virtual disk. Suspending a replication relationship is particularly useful when you want to run a backup of the data on the secondary virtual disk.

When a replication relationship is in a suspended state, the primary virtual disk does not make any attempt to contact the secondary virtual disk. Any writes to the primary virtual disk are persistently logged in the replication repository virtual disks. After the replication relationship resumes, any data that is written to the primary virtual disk is automatically written to the secondary virtual disk. Only the modified data blocks on the primary virtual disk are written to the secondary virtual disk. Full synchronization is not required.

NOTE: If you suspend a remote replication that is set up in the Write consistency mode, you suspend all remote replicated pairs within the group. You can then resume replication operations for any of the individual remote replicated pairs that are in the group.

This example shows the `suspend remoteReplication` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "suspend remoteReplication primary Jan_04_Account
writeConsistency=false;"
```

The `writeConsistency` parameter defines whether the virtual disks identified in this command are in a write-consistency group or are separate. For the virtual disks in a write-consistency group, set this parameter to `TRUE`. For the virtual disks that are not in a write-consistency group, set this parameter to `FALSE`.

This example shows how to use the command in a script file:

```
suspend remoteReplication virtualDisk Jan_04_Account
writeConsistency=false;
```

The replication relationship remains suspended until you use the `resume remoteReplication` command to restart synchronization activities. This command restarts data transfers between a primary virtual disk and a secondary virtual disk in a replication relationship after the replication has been suspended or unsynchronized.

This example shows the `resume remoteReplication` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89
-c "resume remoteReplication virtualDisk Jan_04_Account
writeConsistency=false;"
```

The `writeConsistency` parameter in this command operates the same as in the previous command.

This example shows how to use the command in a script file:

```
resume remoteReplication virtualDisk Jan_04_Account
writeConsistency=false;
```

Removing A Replication Relationship

Use the `remove remoteReplication` command to remove the link between a primary virtual disk and a secondary virtual disk. (Removing a replication relationship is similar to deleting a replication relationship.) Removing the link between a primary virtual disk and a secondary virtual disk does not affect any of the existing data on either virtual disk. The link between the virtual disks is removed, but the primary virtual disk still continues normal I/O operations. Later, you can establish the replication relationship between the two virtual disks and resume normal replication operations. You can remove the replication relationship for one or several remote replicated pairs with this command.

This example shows the `remove remoteReplication` command:

```
c:\...\smX\client>smcli 123.45.67.88 123.45.67.89

-c "remove remoteReplication localVirtualDisk [Jan_04_Account];
```

When you run this command, use the name of the primary virtual disk of the remote replicated pair.

This example shows how to use the command in a script file:

```
remove remoteReplication localVirtualDisk [Jan_04_Account];
```

To re-establish the link between a primary virtual disk and a secondary virtual disk, use the `create remoteReplication` command.

Deleting A Primary Virtual Disk Or A Secondary Virtual Disk

Use the `delete virtualDisk` command to remove a primary virtual disk or a secondary virtual disk from a storage array. Deleting a virtual disk in a replication relationship removes the replication relationship and completely deletes the virtual disk from the storage array. You cannot redefine the replication relationship until you create a new virtual disk or choose an alternate virtual disk to replace the deleted virtual disk.

CAUTION: Possible loss of data access – Deleting a primary virtual disk or a secondary virtual disk permanently removes the data from the storage array.

Disabling The Remote Replication (Legacy) Premium Feature

You disable the Remote Replication (legacy) premium feature to prevent the new replication relationship from being created. When you disable the Remote Replication (legacy) premium feature, the premium feature is in a Disabled/Active state. In this state, you can maintain and manage previously existing replication relationships; however, you cannot create new relationships. To disable the Remote Replication (legacy) premium feature, use this command:

```
disable storageArray feature=remoteReplication
```

Deactivating The Remote Replication (Legacy) Premium Feature

If you no longer require the Remote Replication (legacy) premium feature and you have removed all of the replication relationships, you can deactivate the premium feature. Deactivating the premium feature re-establishes the normal use of dedicated ports on both storage arrays and deletes both replication repository virtual disks. To deactivate the Remote Replication (legacy) premium feature, use this command:

```
deactivate storageArray feature=remoteReplication
```

Interaction With Other Premium Features

You can run the Remote Replication (legacy) premium feature while running these premium features:

- Snapshot—both standard Snapshot and Snapshot (legacy) premium features
- Virtual Disk Copy

When you run the Remote Replication (legacy) premium feature with other premium features, you must consider the requirements of the other premium features to ensure that you set up a stable storage array configuration.

In addition to running with the premium features, you can also run the Remote Replication (legacy) premium feature while running Storage Partitioning and Virtual Disk Expansion (VDE).

Snapshot Virtual Disks

A snapshot virtual disk is a point-in-time image of a virtual disk. It is typically created so that an application, such as a backup, can access the snapshot virtual disk and read the data while the source virtual disk remains online and accessible to hosts.

CAUTION: Before using the source virtual disk of a snapshot virtual disk as your target, you must disable all snapshot virtual disks associated with the source virtual disk. By disabling the snapshot virtual disks, you avoid altering the snapshot data if the source virtual disk is changed.

Creating a snapshot virtual disk automatically creates a snapshot repository virtual disk. The snapshot repository virtual disk stores information about the data that has changed since the snapshot virtual disk

was created. Snapshot repository virtual disks cannot be selected as a source virtual disk or target virtual disk in a virtual disk copy.

The virtual disk for which the point-in-time image is created is the source virtual disk and must be a standard virtual disk in the storage array.

You can select snapshot virtual disks as the source virtual disk for a virtual disk copy. Selecting a snapshot virtual disk is a good use of this feature, because it enables complete backups without significant impact to the storage array I/O. However, some I/O processing resources are lost to the copy operation.

The Snapshot Virtual Disk feature can be used with the Virtual Disk Copy feature to back up data on the same storage array and to restore the data on the snapshot virtual disk back to its original source virtual disk.

Virtual Disk Copy

The Virtual Disk Copy premium feature copies data from one virtual disk (the source virtual disk) to another virtual disk (the target virtual disk) within a single storage array. You can use this premium feature to perform these functions:

- Copy data from disk groups that use smaller-capacity physical disks to disk groups that use larger-capacity physical disks
- Back up data
- Restore snapshot virtual disk data to the base virtual disk.

You can use a primary virtual disk in a remote replication as a source virtual disk or a target virtual disk in a virtual disk copy. You cannot use a secondary virtual disk as a source virtual disk or a target virtual disk.

NOTE: If you start a role reversal during a copy-in-progress, the copy fails and cannot be restarted.

Storage Partitioning

Storage partitioning enables hosts to share access to virtual disks in a storage array. You create a storage partition when you define the following storage array assignments:

- A host
- A host group
- Virtual disk-to-logical unit number (LUN) mapping

The virtual disk-to-LUN mapping enables you to define which host group or host has access to a particular virtual disk in the storage array.

After you create a virtual disk copy, the target virtual disk automatically becomes read-only to hosts to ensure that the data is preserved. Hosts that have been mapped to a target virtual disk do not have write access to the virtual disk, and any attempt to write to the read-only target virtual disk results in a host I/O error.

If you want hosts to have write access to the data on the target virtual disk, use the `set virtualDiskCopy` command to disable the read-only attribute for the target virtual disk.

Virtual Disk Expansion

A Virtual Disk Expansion (VDE) is a modification operation that increases the capacity of a standard virtual disk or a snapshot repository virtual disk. The increase in capacity is achieved by using the free capacity that is available in the disk group of the standard virtual disk or the snapshot repository virtual disk.

This modification operation is considered to be ‘dynamic because you can continually access data on disk groups, virtual disks, and physical disks throughout the entire operation.

A VDE operation can be performed on a primary virtual disk or a secondary virtual disk of a replication relationship.

You cannot perform a VDE operation on a replication repository virtual disk.

10

Maintaining A Storage Array

Maintenance covers a broad spectrum of activities. Its goal is to keep a storage array operational and available to all hosts. This chapter provides descriptions of command line interface (CLI) and script commands that you can use to perform storage array maintenance. The commands are organized into four sections:

- Routine maintenance
- Performance tuning
- Troubleshooting and diagnostics
- Recovery operations

The organization is not a rigid approach, and you can use the commands as appropriate for your storage array. The commands listed in this chapter do not cover the entire array of commands you can use for maintenance. Other commands, particularly the **set** commands, can provide diagnostic or maintenance capabilities.

Routine Maintenance

Routine maintenance involves those tasks you might perform periodically to ensure that the storage array is running as well as possible or to detect conditions before they become problems.

Running A Media Scan

Media scan provides a method of detecting physical disk media errors before they are found during a normal read from or write to the physical disks. Any errors detected are reported to the Major Event Log (MEL). Media scan provides an early indication of a potential drive failure and reduces the possibility of encountering a media error during host operations. A media scan is performed as a background operation and scans all data and consistency information in defined user virtual disks or snapshot groups.

A media scan runs on all virtual disks in the storage array with the following conditions:

- An Optimal status
- No modification operations in progress
- Media scan enabled
- Errors detected during a scan of a user virtual disk are reported to the MEL and handled as:

Unrecovered media error	The physical disk could not read the requested data on its first attempt or on any subsequent retries. For virtual disks with redundancy protection, the data could not be reconstructed from the redundant copy. The error is not corrected but it is reported to the MEL.
Reconstructed media error	The physical disk could not read the requested data on its first attempt or on any subsequent retries. The data is reconstructed from the redundant copy, rewritten to the drive, verified, and the error is reported to the MEL.

Recovered media error	The physical disk could not read the requested data on its first attempt. The result of this action is that the data is rewritten to the physical disk and verified. The error is reported to the MEL.
Consistency mismatches	Consistency errors are found, and a media error is forced on the block stripe so that it is found when the physical disk is scanned again. If consistency is repaired, this forced media error is removed. The result of this action is that the first ten consistency mismatches found on a virtual disk are reported to the MEL.
Unfixable error	The data could not be read and consistency information could not be used to regenerate it. For example, consistency information cannot be used to reconstruct data on a degraded virtual disk. The result of this action is that the error is reported to the MEL.

The script command set provides four commands to define media scan properties:

- `set virtualDisks`
- `set storageArray`
- `set snapGroup`
- `set snapVirtualDisk`

The `set virtualDisk` command enables a media scan for the virtual disk. The following syntax is the general form of the command:

```
set (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNamen] | virtualDisk <wwid>
mediaScanEnabled=(TRUE | FALSE)
```

The `set storageArray` command defines how frequently a media scan is run on a storage array. The following syntax is the general form of the command:

```
set storageArray mediaScanRate=(disabled | 1-30)
```

The `set snapVirtualDisk` and `set snapGroup` commands run media scans on physical disks that are members of snapshots and snapshot groups. The following examples show command syntax:

```
set snapVirtualDisk ["snapvirtualDiskName"]
mediaScanEnabled=(TRUE | FALSE)

set snapGroup ["snapGroupName"]
mediaScanEnabled=(TRUE | FALSE)
```

Running A Consistency Check With A Media Scan

Consistency checks are performed when media scans are run, if consistency check is enabled on the virtual disk. During a consistency check, all data blocks in a virtual disk are scanned, and deteriorated data is corrected. The method of correction depends on the redundant array of independent disks (RAID) levels:

RAID 5 and RAID 6 virtual disks	Consistency is checked and repaired.
RAID 1 virtual disks	The data is compared between the replicated physical disks, and data inconsistencies are repaired.

RAID 0 virtual disks	No redundancy exists. NOTE: Do not run consistency checking commands on RAID 0 virtual disks.

Before attempting a consistency check, you must enable consistency checking using the `set virtualDisk` command. Use the following command to set the `consistencyCheckEnabled` parameter to `TRUE`:

```
set (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNameN] | virtualDisk <wwid>
consistencyCheckEnabled=TRUE
```

Once you have enabled both media scan and consistency check, use the `check virtualDisk consistency` command to write the results to a file (specified by the `mediaErrorFile` and `consistencyErrorFile` parameters):

```
check virtualDisk [virtualDiskName] consistency
[consistencyErrorFile=filename]
[mediaErrorFile=filename]
```

Resetting A RAID Controller Module

CAUTION: When you reset a RAID controller module, the RAID controller module is not available for I/O operations until the reset is complete. If a host is using virtual disks owned by the RAID controller module being reset, the I/O directed to the RAID controller module is rejected. Before resetting the RAID controller module, ensure that a multipath driver is installed on all hosts using these virtual disks. If a multipath driver is not installed, the virtual disks are not available.

Resetting a RAID controller module is the same as rebooting the RAID controller module processors. To reset a RAID controller module, run the following command:

```
reset controller [(0 | 1)]
```

Enabling RAID Controller Module Data Transfer

At times, a RAID controller module might become quiescent while running diagnostics. If this occurs, the RAID controller module might become unresponsive. To revive a RAID controller module, run the following command:

```
enable controller [(0 | 1)] dataTransfer
```

Resetting Battery Age

NOTE: A smart battery module does not require the battery age to be reset.

After replacing the batteries in the storage array, you must reset the age of the battery, either for an entire storage array or one battery in a specific RAID controller module. To reset the age to zero days, run the following command:

```
reset storageArray batteryInstallDate controller=(0 | 1)
```

Removing Persistent Reservations

Persistent reservations preserve virtual disk registrations and prevent hosts, other than the host defined for the virtual disk, from accessing the virtual disk. You must remove persistent reservations before you perform the following changes to your configuration:

- Change or delete logical unit number (LUN) mappings on a virtual disk holding a reservation.
- Delete virtual disk groups or virtual disks that have any reservations.

To determine which virtual disks have reservations, run the following command:

```
show (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNameN]) reservations
```

To clear persistent virtual disk reservations, run the following command:

```
clear (allVirtualDisks |
virtualDisk[virtualDiskName] | virtualDisks
[virtualDiskName1" ... "virtualDiskNameN])reservations
```

Synchronizing RAID Controller Module Clocks

To synchronize the clocks on both RAID controller modules in a storage array with the host clock, run the following command:

```
set storageArray time
```

Locating Physical Disks

At times, you might need to locate a specific physical disk. In very large storage array configurations, this can sometimes be awkward. If you need to locate a specific physical disk, turn on the indicator LED on the front of the physical disk. To turn on the indicator LED on a physical disk, run the following command:

Performance Tuning

Over time, as a storage array exchanges data between the hosts and physical disks, its performance can degrade. Monitor the performance of a storage array and make adjustments to the storage array operational settings to improve performance.

Monitoring Performance

Monitor the performance of a storage array by using the `save storageArray performanceStats` command. This command saves performance information to a file that you can review to determine how well the storage array is running. The following table lists the performance information saved to the file.

Table 14. Storage Array Performance Information

Type of Information	Description	
Devices	Devices are:	
	RAID Controller Modules	The RAID controller module in slot 0 or 1 and a list of the virtual disks owned by the RAID controller module.
	Virtual Disk	A list of the virtual disk names.
	Storage Array Totals	A list of the totals for both RAID controller modules in an active-active RAID controller

Type of Information	Description
	module pair, regardless if one, both, or neither are selected for monitoring
Total I/Os	Number of total I/Os performed since the storage array was started
Read Percentage	Percentage of total I/Os that are read operations (calculate the write percentage by subtracting the read percentage from 100 percent)
Cache Hit Percentage	Percentage of reads that are fulfilled by data from the cache rather than requiring an actual read from a physical disk
Current KB/second	Current transfer rate in kilobytes per second (current means the number of kilobytes per second since the last time the polling interval elapsed, causing an update to occur)
Maximum KB/second	Highest data transfer value achieved in the current kilobyte-per-second statistic block
Current IO/second	Current number of I/Os per second (current means the number of I/Os per second since the last time the polling interval elapsed, causing an update to occur)
Maximum IO/second	Highest number of I/Os achieved in the current I/O-per-second statistic block

The general form of the command is:

```
save storageArray performanceStats file="filename"
```

where, `file` is the name of the file in which you want to save the performance statistics. You can use any file name your operating system can support. The default file type is **.csv**. The performance information is saved as a comma-delimited file.

Before using the `save storageArray performanceStats` command, run the `set session performanceMonitorInterval` and `set session performanceMonitorIterations` commands to specify how often statistics are collected.

Changing RAID Levels

When creating a disk group, define the RAID level for the virtual disks in that group. You can later change the RAID level to improve performance or provide more secure protection for your data. To change the RAID level, run the following command:

```
set diskGroup [diskGroupNumber] raidLevel=(0|1|5|6)
```

where, `diskGroupNumber` is the number of the disk group for which to change the RAID level.

Changing Segment Size

When creating a new virtual disk, define the segment size for that virtual disk. You can later change the segment size to optimize performance. In a multi-user database or file system storage environment, set your segment size to minimize the number of physical disks needed to satisfy an I/O request. Use larger values for the segment size. Using a single physical disk for a single request leaves other disks available to simultaneously service other requests. If the virtual disk is in a single-user large I/O environment,

performance is maximized when a single I/O request is serviced with a single data stripe; use smaller values for the segment size. To change the segment size, run the following command:

```
set virtualDisk ([virtualDiskName] | <wwid>)
segmentSize=segmentSizeValue
```

where, *segmentSizeValue* is the new segment size you want to set. Valid segment size values are `8`, `16`, `32`, `64`, `128`, `256`, and `512`. You can identify the virtual disk by name or World Wide Identifier (WWID) (see Set Virtual Disk).

Defragmenting A Disk Group

When you defragment a disk group, you consolidate the free capacity in the disk group into one contiguous area. Defragmentation does not change the way in which the data is stored on the virtual disks. As an example, consider a disk group with five virtual disks. If you delete virtual disks 1 and 3, your disk group is configured in the following manner:

space, virtual disk 2, space, virtual disk 4, virtual disk 5, original unused space

When you defragment this group, the space (free capacity) is consolidated into one contiguous location after the virtual disks. After being defragmented, the disk group is:

virtual disk 2, virtual disk 4, virtual disk 5, consolidated unused space

To defragment a disk group, run the following command:

```
start diskGroup [diskGroupNumber] defragment
```

where, `diskGroupNumber` is the identifier for the disk group.

NOTE: Defragmenting a disk group starts a long-running operation.

Using SSD Cache

If you are using solid-state disks (SSDs) in your storage array, you can configure SSD cache and model potential workload throughput improvements, including cache hit percentage and average response time. SSD cache improves read-only performance by logically grouping SSDs together to provide secondary cache to the primary cache in the RAID controller module memory.

Enabling SSD Cache For A Virtual Disk

This `set virtualDisk` command turns SSD caching on or off on a standard virtual disk, snapshot virtual disk or a consistency group snapshot virtual disk. Set the `ssdCacheEnabled` parameter to `TRUE` to enable SSD caching.

For standard virtual disks:

```
set virtualDisk ["virtualdiskName"]
ssdCacheEnabled=TRUE
```

For snapshot virtual disks:

```
set snapVirtualDisk ["virtualdiskName"]
ssdCacheEnabled=TRUE
```

For a consistency group snapshot virtual disk:

```
set cgSnapVirtualDisk ["cgSnapshotVirtualDisk"]
ssdCacheEnabled=TRUE
```

When you turn off SSD caching for a virtual disk, cache contents is purged. You can turn on SSD caching for only one virtual disk per command.

Creating SSD Cache

To create `ssdCache` command SSD read cache:

```
create ssdCache userLabel="ssdCacheName"
physicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
[updateExistingVirtualDisks=(TRUE|FALSE)]
```

The `ssdCacheName` parameter names the cache group. Setting the `updateExistingVirtualDisks` parameter to `TRUE` enables SSD caching for all virtual disks in the storage array.

The following restrictions apply:

- A storage array can have only one SSD cache.
- Only virtual disks created using physical disks can use the SSD cache. You cannot enable SSD cache on snapshot (legacy) virtual disks or snapshot images.
- SSD caching does not support self-encrypting disks (SEDs).

Adding Physical Disks To Existing SSD Cache

To add to or increase the capacity of an existing SSD cache, use the `set ssdCache` command:

```
set ssdCache [ssdCacheName]
addPhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
```

Removing Physical Disks From Existing SSD Cache

To remove physical disks from existing SSD Cache:

```
set ssdCache [ssdCacheName]
removePhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
```

You cannot remove all physical disks from an SSD Cache, and at least one physical disk must remain. To completely remove the SSD Cache, use the `delete ssdCache` [*ssdCacheName*] command.

SSD Cache Performance Modeling

SSD Cache Performance Modeling helps you determine potential performance improvement based on different SSD cache capacities. The modeling tool provides an estimate of performance using the following criteria:

- cache hit percentage
- average response time

To start the modeling tool from the command line:

```
start ssdCache [ssdCacheName]
performanceModeling
```

The modeling process runs in the background until you run the stop command. The `file` parameter is optional. If used, it will write modeling results to the file you specify.

```
stop ssdCache [ssdCacheName]
performanceModeling file="filename"
```

Output is displayed directly on your screen, unless the `file` parameter is used. You can also access modeling results using MD Storage Manager.

NOTE: If you reboot a RAID controller module or make configuration changes to any SSD Cache options during the modeling run, no data will be saved.

Additional Information About SSD Cache And Performance Modeling

For complete description of the SSD Cache feature, see the *Administrator's Guide*.

The following additional SSD cache-related commands are available in this document:

- `Change SSD Cache Application Type`
- `Blink SSD Cache`
- `Rename SSD Cache`
- `Resume SSD Cache`
- `Show SSD Cache`
- `Suspend SSD Cache`

Troubleshooting and Diagnostics

If a storage array exhibits abnormal operation or failures, you can use the commands described in this section to determine the cause of the problems.

Collecting Physical Disk Data

To gather information about all the physical disks in a storage array, run the `save allPhysicalDisks` command. This command collects sense data from all the physical disks in a storage array and saves the data to a file. The sense data consists of statistical information maintained by each of the physical disks in the storage array.

Diagnosing A RAID Controller Module

The `diagnose controller` command's `testID` parameter takes the following options, which you can use to verify that a RAID controller module is functioning correctly:

1	Reads the test
2	Performs a data loop-back test
3	Writes the test

The read test initiates a `read` command as it would be sent over an I/O data path. The read test compares data with a known, specific data pattern, checking for data integrity and errors. If the `read` command is unsuccessful or the data compared is not correct, the RAID controller module is considered to be in error and is placed offline.

Run the data loopback test only on RAID controller modules that have connections between the RAID controller module and the physical disks. The test passes data through each RAID controller module

physical disk-side channel out onto the loop and back again. Enough data is transferred to determine error conditions on the channel. If the test fails on any channel, this status is saved so that it can be returned if all other tests pass.

The write test initiates a `write` command as it would be sent over an I/O data path to the diagnostics region on a specified physical disk. This diagnostics region is then read and compared to a specific data pattern. If the write fails or the data compared is not correct, the RAID controller module is considered to be in error, and it is failed and placed offline.

For best results, run all three tests at initial installation. Also, run the tests any time you make changes to the storage array or to components connected to the storage array (such as hubs, switches, and host adapters).

A custom data pattern file called **diagnosticsDataPattern.dpf** is included on the **Utility** directory of the installation CD. You can modify this file, but the file must have the following properties to work correctly for the tests:

- The file values must be entered in hexadecimal format (00 to FF) with only one space between the values.
- The file must be no larger than 64 bytes in size. Smaller files can be used, but larger files can cause an error.

The test results contain a generic, overall status message and a set of specific test results. Each test result contains the following information:

- Test (read/write/data loopback)
- Port (read/write)
- Level (internal/external)
- Status (pass/fail)

Events are written to the MEL when diagnostics are started and when testing is completed. These events help you to evaluate whether diagnostics testing was successful or failed and the reason for the failure.

Recovery Operations

Recovery operations involve replacing failed RAID controller modules and physical disks, restoring data, and restoring the storage array to operation.

Setting RAID Controller Module Operational Mode

A RAID controller module has three operational modes:

- Online
- Offline
- Service

CAUTION: Placing a RAID controller module offline can cause loss of data.

Placing a RAID controller module online sets it to the Optimal state and makes it active and available for I/O operations. Placing a RAID controller module offline makes it unavailable for I/O operations and moves its disk groups to the other RAID controller module if failover protection is enabled.

Taking a RAID controller module offline can seriously impact data integrity and storage array operation.

If you take a RAID controller module offline, the second RAID controller module in the pair takes over. Disk groups and their associated virtual disks that were assigned to the offline RAID controller module are automatically reassigned to the remaining RAID controller module.

CAUTION: Place a RAID controller module in Service mode only under the direction of Technical Support.

Use Service mode when you want to perform an operation, such as replacing a RAID controller module. Placing a RAID controller module in Service mode makes it unavailable for I/O operations. Placing a RAID controller module in Service mode also moves the disk groups from the RAID controller module to the second RAID controller module without affecting the disk groups' preferred path. Moving disk groups might significantly reduce performance. The disk groups are automatically transferred back to the preferred RAID controller module when it is placed back online.

CAUTION: A multipath driver is required on all hosts and is the only supported configuration. If the multipath driver is not installed, the virtual disks are not accessible.

Before you place a RAID controller module in Service mode, ensure that a multipath driver is installed on all hosts using these virtual disks.

To change the operational mode of a RAID controller module, run the following command:

```
set controller [(0 | 1)] availability=(online | offline | serviceMode)
```

Changing RAID Controller Module Ownership

You can change which RAID controller module owns a virtual disk by using the `set virtualDisk` command. The following syntax is the general form of the command:

```
set (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNameN] | virtualDisk <wwid> owner=(0 | 1)
```

Initializing A Physical Disk

CAUTION: When you initialize a physical disk, all data on the physical disk is lost.

You must initialize a physical disk when you have moved physical disks that were previously part of a disk group from one storage array to another. If you do not move the entire set of physical disks, the disk group and virtual disk information on the physical disks that you move is incomplete. Each physical disk that you move contains only part of the information defined for the virtual disk and disk group. To be able to reuse the physical disks to create a new disk group and virtual disk, you must erase all old information from the physical disks by initializing the physical disk.

When you initialize a physical disk, all old disk group and virtual disk information is erased, and the physical disk is returned to an unassigned state. Returning a physical disk to an unassigned state adds unconfigured capacity to a storage array. You can use this capacity to create additional disk groups and virtual disks.

To initialize a physical disk, run the following command:

```
start physicalDisk [enclosureID,slotID] initialize
```

where, *enclosureID* and *slotID* are the identifiers for the physical disk.

Reconstructing A Physical Disk

If two or more physical disks in a disk group have failed, the virtual disk shows a status of Failed. All of the virtual disks in the disk group are no longer operating. To return the disk group to an Optimal status, you must replace the failed physical disks. After replacing the physical disks, reconstruct the data on physical disks. The reconstructed data is the data as it would appear on the failed physical disks.

To reconstruct a physical disk, run the following command:

```
start physicalDisk [enclosureID,slotID] reconstruct
```

where, *enclosureID* and *slotID* are the identifiers for the physical disk.

NOTE: You can use this command only when the physical disk is assigned to a RAID 1, 5, or 6 disk group.

Initializing A Virtual Disk

CAUTION: When you initialize a virtual disk, all data on the virtual disk and all information about the virtual disk are erased.

A virtual disk is automatically initialized when you first create it. If the virtual disk starts exhibiting failures, you might be required to re-initialize the virtual disk to correct the failure condition.

The initialization process cannot be cancelled once it has begun. This option cannot be used if any modification operations are in progress on the virtual disk or disk group. To initialize a virtual disk, run the following command:

```
start virtualDisk [virtualDiskName] initialize
```

where, *virtualDiskName* is the identifier for the virtual disk.

NOTE: Initializing a thin virtual disk may take an extremely long time.

Redistributing Virtual Disks

Redistributing virtual disks returns the virtual disks to their preferred RAID controller module owners. The preferred RAID controller module ownership of a virtual disk or disk group is the RAID controller module of an active-active pair that is designated to own the virtual disks. The preferred owner for a virtual disk is initially designated when the virtual disk is created. If the preferred RAID controller module is being replaced or undergoing a firmware download, ownership of the virtual disks is automatically shifted to the second RAID controller module. The second RAID controller module becomes the current owner of the virtual disks. This change is considered to be a routine ownership change and is reported in the MEL.

CAUTION: Ensure that a multipath driver is installed, or the virtual disks are not accessible.

To redistribute virtual disks to their preferred RAID controller modules, run the following command:

```
reset storageArray virtualDiskDistribution
```

NOTE: You cannot run this command if all virtual disks are currently owned by their preferred RAID controller module or if the storage array does not have defined virtual disks.

Under some host operating systems, you must reconfigure the multipath host driver. You might also need to make operating system modifications to recognize the new I/O path to the virtual disk.

11

Script Commands

CAUTION: Script commands are capable of changing the configuration and may cause loss of data if not used correctly. Command operations are performed as soon as you run the commands. Before using the script commands, ensure that you have backed up all data, and have saved the current configuration so that you can reinstall it if the changes do not work.

This chapter describes the script commands used to configure, monitor, and maintain a storage array. This chapter is organized into four sections:

- Command Formatting Rules lists general formatting rules that apply to the command syntax.
- Firmware Compatibility Levels lists the firmware level for this release.
- Commands Listed by Function lists the commands by functional activity:
  - Asynchronous Remote Replication
  - Disk group
  - Enclosure
  - Host topology
  - Physical disk
  - Redundant array of independent disks (RAID) controller module
  - Remote Replication (legacy)
  - Session
  - Show string
  - Snapshot
  - Storage array
  - Virtual disk
  - Virtual disk copy
- Commands Listed Alphabetically lists the commands alphabetically and, for each command, includes command name, syntax, and parameters.

CAUTION: Commands entered using the command line interface (CLI) are capable of damaging a configuration and causing loss of data if not used properly. Command operations are performed as soon as you run the commands. Some commands can immediately delete configurations or data. Before using the command line interface, make sure you have backed up all data, and save the current configuration so that you can reinstall it if the changes you make do not work.

Command Formatting Rules

This section describes the general rules for formatting a script command and how the command syntax is presented in the following command descriptions. Syntax unique to a specific command is explained in the notes at the end of each command description.

- The script commands are not case sensitive. Type the commands in lowercase, uppercase, or mixed case. (In the following command descriptions, mixed case is used as an aid to reading the command names and understanding the purpose of the command.)
- You must enter spaces in the commands as they are shown in the command descriptions.
- Brackets are used in two ways:
  - As part of the command syntax
  - To indicate which parameters are optional

  The description of each parameter tells you when you must put brackets around a parameter value.
- Parentheses shown in the command syntax enclose specific choices for a parameter. That is, if you want to use the parameter, you must use one of the values shown in the parentheses. Generally, you do not include parentheses in a command. In some instances, however, you must put parentheses around a list. For example, you must put parentheses around a list of enclosure ID values and slot ID values. The description of each parameter tells you if you must put parentheses around a parameter value.
- Vertical bars in a command indicate or and separate the valid entries for the parameter. For example, the syntax for the `raidLevel` parameter in the command description appears as follows:

  ```
  raidLevel=(0 | 1 | 5 | 6)
  ```

  To use the `raidLevel` parameter to set a RAID level of 5, enter:

  ```
  raidLevel=5
  ```
- When you specify physical disk locations by using enclosure ID values and slot ID values, separate the ID values with a comma. If you enter more than one set of ID values, separate each set of values by a space. Put parentheses around the set of values. For example:

  ```
  (0,0 0,1 0,2 0,3 1,0 1,1 1,2 1,3)
  ```
- Italicized terms in the command indicate a value or information that you need to provide. For example, when you encounter the italicized term:

  numberOfPhysicalDisks

  Replace the italicized term with a value for the number of physical disks that you want to include with the command.
- You can use any combination of alphanumeric characters, hyphens, and underscores for the names of the following components:
  - Storage arrays
  - Host groups
  - Hosts
  - Disk groups
  - Snapshot groups
  - Snapshot images
  - Virtual disks
  - Host bus adapter (HBA) host ports

  Names can have a maximum of 30 characters. If the label contains multiple words, underscores, or hyphens, you must put quotation marks around the name. In some usages you must also put brackets around the name. The description of each parameter tells you if you must put quotation marks or brackets around a parameter value. The character string cannot contain a new line. You must use unique names or the RAID controller module firmware returns an error.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

On Microsoft Windows, you must put a backslash (\) before and after the name in addition to other delimiters. For example, the following name is used in a command running under a Windows operating system:

```
[\"Engineering"\]
```

For Linux, and when used in a script file, the name appears as the following:

```
["Engineering"]
```

When you enter the World Wide Identifier (WWID) of an HBA host port, some usages require quotation marks around the WWID. In other uses, you must put angle brackets (< >) around the WWID. The description of the WWID parameter tells you if you must put quotation marks or angle brackets around the WWID.

- Script commands must end with a semicolon (;). You can enter more than one script command on the command line each time you enter a CLI command.

Firmware Compatibility Levels

The supported RAID controller firmware for SMcli commands is 7.70.xx.xx, and later.

NOTE:

Modular Disk Storage Manager (MDSM) is backward compatible. When the latest Resource DVD is used to install MDSM, all SMcli commands will work with the supported controller firmware.

Commands Listed By Function

This section presents the commands organized into groups related to physical, logical, and operational features of the storage array.

Disk Group Commands

Create Disk Group

Delete Disk Group

Enable Disk Group Security

Resume Snapshot Rollback

Set Disk Group

Troubleshooting

Show Disk Group

Show Disk Group Export Dependencies

Show Disk Group Import Dependencies

Start Disk Group Blink

Start Disk Group Defragment

Start Disk Group Export

Start Disk Group Import

Stop Disk Group Blink

Enclosure Commands

Download Enclosure Management Module Firmware

Save Enclosure Log Data

Set Enclosure Attribute

Set Enclosure Identification

Start Enclosure Blink

Stop Enclosure Blink

Host Topology Commands

Create Host

Create Host Group

Create Host Port

Create iSCSI Initiator

Delete Host

Delete Host Group

Delete Host Port

Delete iSCSI Initiator

Set Host

Set Host Group

Set Host Channel

Set Host Port

Set iSCSI Initiator

Set iSCSI Target Properties

Show Host Ports

iSCSI Commands

Create iSCSI Initiator

Delete iSCSI Initiator

Reset Storage Array iSCSI Baseline

Save Storage Array iSCSI Statistics

Set iSCSI Initiator

Set iSCSI Target Properties

Show Current iSCSI Sessions

Show Storage Array Negotiation Defaults

Start iSCSI DHCP Refresh

Stop Storage Array iSCSI Session

Physical Disk Commands

Clear Physical Disk Port Statistics

Download Physical Disk Firmware

Replace Physical Disk

Revive Physical Disk

Save Physical Disk Channel Fault Isolation Diagnostic Status

Save Physical Disk Log

Set Virtual Disk

Set Foreign Physical Disk to Native

Set Physical Disk Hot Spare

Set Physical Disk Channel Status

Set Physical Disk State

Show Physical Disk

Show Physical Disk Channel Statistics

Show Physical Disk Download Progress

Start Physical Disk Blink

Start Physical Disk Initialize

Start Physical Disk Channel Blink

Start Physical Disk Channel Fault Isolation Diagnostics

Start Physical Disk Reconstruction

Start Remote Replication Synchronization

Stop Physical Disk Blink

Stop physical disk channel blink

Stop Physical Disk Channel Fault Isolation Diagnostics

RAID Controller Module Commands

Clear Physical Disk Port Statistics

Diagnose RAID Controller Module

Enable RAID Controller Module Data Transfer

Reset RAID Controller Module

Save RAID Controller Module NVSRAM

Set Host Channel

Set Host Port

Set Physical Disk Channel Status

Set RAID Controller Module

Show Physical Disk Channel Statistics

Show RAID Controller Module

Show RAID Controller Module NVSRAM

Start Physical Disk Channel Blink

Start Physical Disk Channel Fault Isolation Diagnostics

Stop Physical Disk Channel Blink

Stop Physical Disk Channel Fault Isolation Diagnostics

Remote Replication Commands

Activate Remote Replication

Add Virtual Disk to Remote Replication Group

Cancel Remote Replication Group Role Reversal

Check Remote Replication Group Consistency

Check Remote Replication Status

Create Remote Replication

Check Storage Array Connectivity

Clear Remote Replication Fault

Deactivate Remote Replication

Deactivate Remote Replication (Legacy)

Delete Remote Replication Group

Diagnose RAID Controller Module iSCSI Host Cable

Establish Remote Replicated Pair

Re-Create Remote Replication Repository Virtual Disk

Remove Incomplete Remote Replicated Pair from Remote Replication Group

Remove Remote Replication

Remove Virtual Disk from Remote Replication Group

Reset iSCSI IP Address

Reset Remote Replication Group Statistics

Resume Remote Replication

Resume Remote Replication Group

Save Remote Replication Group Statistics

Set Remote Replication

Set Remote Replication Group

Show Remote Replication Group

Show Remote Replication Group Synchronization Progress

Show Remote Replication Virtual Disk Candidates

Show Remote Replication Virtual Disk Synchronization Progress

Start Remote Replication Synchronization

Start Snapshot Rollback

Suspend Remote Replication

Suspend Remote Replication Group

Test Remote Replication Group Connectivity

Session Command

Set Session

Show String Command

Show String

Snapshot Commands

Create Snapshot Group

Create Snapshot Image

Create Snapshot Virtual Disk

Re-Create Snapshot

Set Snapshot Virtual Disk

Stop Snapshot

SSD Cache Commands

Add Physical Disks to SSD Cache

Change SSD Cache Application Type

Create SSD Cache

Delete SSD Cache

Enable or Disable SSD Cache for a Virtual Disk

Blink SSD Cache

Remove Physical Disks from SSD Cache

Rename SSD Cache

Resume SSD Cache

Show SSD Cache

Show SSD Cache Statistics

Start SSD Cache Performance Modeling

Stop SSD Cache Performance Modeling

Suspend SSD Cache

Storage Array Commands

Activate Storage Array Firmware

Autoconfigure Storage Array

Autoconfigure Storage Array Hot Spares

Clear Storage Array Configuration

Clear Storage Array Event Log

Clear Storage Array Firmware Pending Area

Scheduling Snapshots

Disable Storage Array Feature

Download Storage Array Firmware/NVSRAM

Enable RAID Controller Module Data Transfer

Download Storage Array NVSRAM

Enable Storage Array Feature

Reset Storage Array Battery Install Date

Reset Storage Array Battery Install Date

Reset Storage Array SAS PHY Baseline

Reset Storage Array Virtual Disk Distribution

Save Storage Array Configuration

Save Storage Array Events

Save Storage Array iSCSI Statistics

Save Storage Array Performance Statistics

Save Storage Array SAS PHY Counts

Save Storage Array State Capture

Save Storage Array Support Data

Set Storage Array

Set Storage Array Enclosure Positions

Set Storage Array ICMP Response

Set Storage Array iSNS Server IPv4 Address

Set Storage Array iSNS Server IPv6 Address

Set Storage Array iSNS Server Listening Port

Set Storage Array Learn Cycle

Set Storage Array Redundancy Mode

Set Storage Array Security Key

Set Storage Array Time

Set Storage Array Unnamed Discovery Session

Show Storage Array

Show Storage Array Auto Configure

Show Storage Array Host Topology

Show Storage Array LUN Mappings

Show Storage Array Negotiation Defaults

Show Storage Array Unconfigured iSCSI Initiators

Show Storage Array Unreadable Sectors

Stop Snapshot Rollback

Start Storage Array iSCSI Refresh

Stop Storage Array Blink

Stop Storage Array Physical Disk Firmware Download

Validate Storage Array Security Key

Virtual Disk Commands

Check Virtual Disk Consistency

Clear Virtual Disk Reservations

Clear Virtual Disk Unreadable Sectors

Create RAID Virtual Disk (Automatic Physical Disk Select)

Create RAID Virtual Disk (Free Capacity Base Select)

Create RAID Virtual Disk (Manual Physical Disk Select)

Delete Virtual Disk

Remove Virtual Disk LUN Mapping

Repair Virtual Disk Consistency

Set Virtual Disk

Show Virtual Disk

Show Virtual Disk Action Progress

Show Virtual Disk Performance Statistics

Show Virtual Disk Reservations

Start Virtual Disk Initialization

Virtual Disk Copy Commands

Create Virtual Disk Copy

Recopy Virtual Disk Copy

Remove Virtual Disk Copy

Set Virtual Disk Copy

Show Virtual Disk Copy

Show Virtual Disk Copy Source Candidates

Show Virtual Disk Copy Target Candidates

Stop Virtual Disk Copy

12

Commands Listed Alphabetically

Following are the script commands listed alphabetically.

Activate Remote Replication Feature

NOTE: This command is used to activate both Remote Replication and Remote Replication (Legacy).

Description

This command creates the replication repository virtual disks and activates the Remote Replication and Remote Replication (legacy) premium features. When using this command, the replication repository virtual disk can be defined in one of three ways:

- User-defined physical disks
- User-defined disk group
- User-defined number of physical disks

If you define a number of physical disks, the RAID controller module firmware selects which physical disks to use for the replication repository virtual disk.

Syntax

User-Defined Physical Disks

```
activate storageArray feature=remoteReplication
repositoryRAIDLevel=(0 |1 | 5 | 6)
repositoryPhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn) [repositoryDiskGroupUserLabel="diskGroupName"]
[enclosureLossProtect=(TRUE | FALSE)]
[drawerLossProtect=(TRUE | FALSE)]
[dataAssurance=(none | enabled)]
```

User-Defined Disk Group

```
activate storageArray feature=remoteReplication
repositoryDiskGroup=diskGroupName
[freeCapacityArea=freeCapacityIndexNumber]
```

User-Defined Number Of Physical Disks

```
activate storageArray feature=remoteReplication
repositoryRAIDLevel=(1 | 5 | 6)
repositoryPhysicalDiskCount=numberOfPhysicalDisks
[repositoryDiskGroupUserLabel="diskGroupName"]
[enclosureLossProtect=(TRUE | FALSE)]
```

```
[drawerLossProtect=(TRUE | FALSE)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`repositoryRAIDLevel`	The RAID level for the replication repository virtual disk. Valid values are **0**, **1**, **5**, or **6**.
`repositoryPhysicalDisks`	The physical disks for the replication repository virtual disk. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each physical disk that you assign to the replication repository virtual disk. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each physical disk that you assign to the replication repository virtual disk. Enclosure ID values are **0** to **99**. Drawer ID values are **0** to **4**. Slot ID values are **0** to **31**. Enclose the enclosure ID values, the drawer ID values, and the slot ID values in parentheses.
`repositoryDiskGroupUserLabel`	The name that you want to give the new disk group in which the replication repository virtual disk is located. Enclose the disk group name in double quotation marks (" ").
`repositoryDiskGroup`	The name of the replication repository disk group where the replication repository virtual disk is located. To determine the names of the disk groups in your storage array, run the `show storageArray profile` command.
`freeCapacityArea`	The index number of the free space in an existing disk group that you want to use to create the replication repository virtual disk. Free capacity is defined as the free capacity between existing virtual disks in a disk group. For example, a disk group might have the following areas: virtual disk 1, free capacity, virtual disk 2, free capacity, virtual disk 3, free capacity. To use the free capacity following virtual disk 2, you would specify: `freeCapacityArea=2` Run the `show diskGroup` command to determine if a free capacity area exists.
`repositoryPhysicalDiskCount`	The number of unassigned physical disks that you want to use for the replication repository virtual disk.

Parameter	Description
`enclosureLossProtect`	The setting to enforce enclosure loss protection when you create the replication repository virtual disk. To enforce enclosure loss protection, set this parameter to **TRUE**. The default value is **FALSE**.
`drawerLossProtect`	The setting to enforce drawer loss protection when you create the replication repository virtual disk. To enforce drawer loss protection, set this parameter to **TRUE**. The default value is **FALSE**.
`dataAssurance`	The setting to specify that a disk group, and the virtual disks within the disk group, has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk group. These settings are valid: • **none** - The disk group does not have data assurance protection. • **enabled** - The disk group has data assurance protection. The disk group supports protected information and is formatted with protection information enabled.

Notes

The `repositoryPhysicalDisks` parameter supports both high-capacity expansion enclosures and low-capacity expansion enclosures. A high-capacity expansion enclosure has drawers that hold the physical disks. The drawers slide out of the expansion enclosure to provide access to the physical disks. A low-capacity expansion enclosure does not have drawers. For a high-capacity expansion enclosure, you must specify the identifier (ID) of the expansion enclosure, the ID of the drawer, and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, you need only specify the ID of the expansion enclosure and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, an alternative method for identifying a location for a physical disk is to specify the ID of the expansion enclosure, set the ID of the drawer to **0**, and specify the ID of the slot in which a physical disk resides.

If the physical disks that you select for the `repositoryPhysicalDisks` parameter are not compatible with other parameters (such as the `repositoryRAIDLevel` parameter), the script command returns an error, and Remote Replication (Legacy) is not activated. The error returns the amount of space that is needed for the replication repository virtual disk. You can then re-enter the command, and specify the appropriate amount of space.

If you enter a value for the repository storage space that is too small for the replication repository virtual disks, the RAID controller module firmware returns an error message that provides the amount of space that is needed for the replication repository virtual disks. The command does not try to activate Remote Replication (Legacy). You can re-enter the command by using the value from the error message for the repository storage space value.

When you assign the physical disks, if you set the `enclosureLossProtect` parameter to **TRUE** and have selected more than one physical disk from any one enclosure, the storage array returns an error. If you

set the `enclosureLossProtect` parameter to **`FALSE`**, the storage array performs operations, but the disk group that you create might not have enclosure loss protection.

When the RAID controller module firmware assigns the physical disks, if you set the `enclosureLossProtect` parameter to **`TRUE`**, the storage array returns an error if the RAID controller module firmware cannot provide physical disks that result in the new disk group having enclosure loss protection. If you set the `enclosureLossProtect` parameter to **`FALSE`**, the storage array performs the operation even if it means that the disk group might not have enclosure loss protection.

The `drawerLossProtect` parameter defines if data on a virtual disk is accessible if a drawer fails. When you assign the physical disks, if you set the `drawerLossProtect` parameter to **TRUE** and select more than one physical disk from any one drawer, the storage array returns an error. If you set the `drawerLossProtect` parameter to **`FALSE`**, the storage array performs operations, but the disk group that you create might not have drawer loss protection.

You must set the `enclosureLossProtect` parameter and the `drawerLossProtect` parameter to the same value. Both of the parameters must be either **`TRUE`** or **`FALSE`**. If the `enclosureLossProtect` parameter and the `drawerLossProtect` parameter is set to different values, the storage array returns an error.

Activate Storage Array Firmware

Description

This command activates firmware previously downloaded to the pending configuration area in the RAID controller modules in the storage array.

Syntax

```
activate storageArray firmware
```

Parameters

None

Add Member To Consistency Group

Description

This command adds a new base virtual disk as a member to a an existing consistency group. You can specify an existing repository virtual disk for the new consistency group member, or create a new repository virtual disk. When you create a new repository virtual disk, you identify an existing disk group or an existing disk pool where you want the repository virtual disk.

Syntax For Use With An Existing Repository Virtual Disk

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName"
repositoryVirtualDisk="repositoryName"
```

Syntax For Use When Creating A New Repository Virtual Disk In A Disk Group

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName"
repositoryVirtualDisk=("diskGroupName" capacity=capacityValue(KB|MB|GB|TB|
bytes))
```

Syntax For Use When Creating A New Repository Virtual Disk In A Disk Pool

```
set consistencyGroup ["consistencyGroupName"]
addCGMemberVirtualDisk="baseVirtualDiskName"
repositoryVirtualDisk=("diskPoolName" capacity=capacityValue(KB|MB|GB|TB|bytes))
```

Parameters

Parameter	Description
`consistencyGroup`	The name of the consistency group to which you want to add a new member virtual disk. The new member virtual disk is the base virtual disk for snapshot operations. Enclose the consistency group name in double quotation marks (" ") inside square brackets ([ ]).
`addCGMemberVirtualDisk`	The name of a base virtual disk that you want to add. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the member name in double quotation marks (" "). If the specified virtual disk is an existing repository virtual disk or an existing snapshot virtual disk, the command fails.
`repositoryVirtualDisk`	This parameter performs two functions: • In an existing consistency group that has a repository virtual disk, this parameter identifies the repository virtual disk. • When creating a new repository virtual disk this parameter identifies either a disk group or disk pool in which to create the new repository virtual disk.
`capacity`	The size of a new repository virtual disk in either a disk group or a disk pool. Size is defined in units of bytes, KB, MB, GB, or TB.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

A consistency group is a collection of base virtual disks that are the source virtual disks for snapshots. You collect the base virtual disks in a consistency group so that you can perform the same snapshot

operations on each of the base virtual disks. In the context of this command, the term *member* means a base virtual disk for snapshot operations. You can manipulate snapshot images associated with members of a consistency group through batch-style operations, while maintaining consistency across the snapshot images.

Each member virtual disk must have a corresponding repository virtual disk. You can define the relationship between the member virtual disk and the repository virtual disk using the `repositoryVirtualDisk` parameter. The `repositoryVirtualDisk` parameter can perform one of these actions:

- Identify an existing repository virtual disk that is connected to the consistency group.
- Identify either a disk group or a disk pool in which you want to create a new repository virtual disk.

Adding a member to a consistency group with an existing repository has two purposes:

- You can create an entirely new repository virtual disk by running the command without the `repositoryVirtualDisk` parameter. When you run the command without the `repositoryVirtualDisk` parameter, the command creates a new repository virtual disk in the disk group or disk pool in which all the other repository virtual disks are stored. An example of this command usage is:

  ```
  set consistencyGroup ["First_Images"] addCGMemberVirtualDisk="Data_020212"
  ```
- You can reuse an existing repository virtual disk if that repository virtual disk is empty and is not related to another member virtual disk. You might want to do this if you want to maintain a particular sequence or relationship for the repository virtual disks. To reuse an existing, empty repository virtual disk you need to know the name of the repository virtual disk. To determine the name of the repository virtual disk, use the `show allVirtualDisks summary` command. All repository virtual disk names have the form **repos_XXXX** where **XXXX** is a unique identifier generated by the MD storage management software. An example of this command usage is:

  ```
  set consistencyGroup ["First_Images"] addCGMemberVirtualDisk="Data_020212"
  repositoryVirtualDisk="repos_0011"
  ```

You can place the repository virtual disk in a disk group or a disk pool of your choosing. You are not required to have the repository virtual disk in the same location as other repository virtual disks. To place the repository virtual disk in a disk group or a disk pool of your choice, use the `repositoryVirtualDisk` parameter and identify the disk group or the disk pool and a size for the repository virtual disk. An example of this command usage is:

```
set consistencyGroup ["First_Images"] addCGMemberVirtualDisk="Data_020212"
repositoryVirtualDisk=("12" capcity=2 GB)
```

In this example, `"12"` is the name of an existing disk group or an existing disk pool. The capacity parameter defines the size that you want for the repository disk group.

When you create a new repository virtual disk in either a disk group or a disk pool, you must include parenthesis around the disk group name and capacity, or the disk pool name and capacity.

Add Physical Disks To SSD Cache

Description

Use this command to increase the capacity of an existing Solid State Disk (SSD) cache by adding additional SSDs.

Syntax

```
set ssdCache [ssdCacheName]
addPhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
```

Parameters

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache to which you want to add SSDs. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
addPhysicalDisks	The physical disks that you want to add to the SSD cache. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each SSD that you want to add. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each SSD that you want to add. Enclosure ID values are 0 to 99. Drawer ID values are 0 to 4. Slot ID values are 0 to 31. Enclose the enclosure ID values, the drawer ID values, and the slot ID values in parentheses.

Notes

The SSD cache can contain any number of SSDs. The maximum size of the SSD cache is 5 TB, but might be less depending on the size of the RAID controller module's primary cache.

Add Virtual Disk To Remote Replication Group

Description

This command adds a primary virtual disk to a remote replication group. This command is valid only on the local storage array that contains the remote replication group to which you want to add the primary virtual disk. A remote replication group has a repository virtual disk that is used to save data for all of the point-in-time images that are part of the remote replication group.

Syntax

```
add virtualDisk="virtualDiskName"
asyncRemoteReplicationGroup="asyncRemoteReplicationGroupName"
remotePassword="password"
(repositoryVirtualDisk="repos_xxxx" | repositoryVirtualDisk=(diskGroupName
[capacity=capacityValue]) repositoryVirtualDisk=(diskPoolName
[capacity=capacityValue]))
```

Parameters

Parameter	Description
`virtualDisk`	The name of the primary virtual disk that you want to add to the remote replication group. Enclose the virtual disk name in double quotation marks (" ").
`asyncRemoteReplicationGroup`	The name of the remote replication group that will contain the member virtual disk that you want to add. Enclose the remote replication group name in double quotation marks (" ").
`remotePassword`	This parameter is optional. Use this parameter when the remote storage array is password protected. Enclose the password in double quotation marks (" ").
`repositoryVirtualDisk`	The name of the repository virtual disk that will contain the remote replication group. You have two options for defining the name of a repository virtual disk: • Use an existing repository virtual disk name. • Create a new repository virtual disk when you run this command. The name of an existing repository virtual disk is comprised of two parts: • The term `repos`. • A four digit numerical identifier that you assign to the repository virtual disk name. Enclose the name of the existing repository virtual disk in double quotation marks (" "). If you want to create a new repository virtual disk when you run this command you must enter the name of either a disk group or a disk pool in which you want the repository virtual disk. Optionally, you also can define the capacity of the repository virtual disk. If you want to define the capacity you can use these values: • An integer value that represents a percentage of the base virtual disk capacity. • A decimal fraction value that represents a percentage of the base virtual disk capacity. • A specific size for the repository virtual disk. Size is defined in units of `bytes`, `KB`, `MB`, `GB`, or `TB`. If you do not use the capacity option, the MD storage management software sets the capacity to 20 percent of the base virtual disk capacity.

Parameter	Description
	Enclose the name of the new repository virtual disk in parenthesis ( ).

Autoconfigure Storage Array

Description

This command automatically configures a storage array. Before entering the `autoConfigure storageArray` command, run the `show storageArray autoConfiguration` command. The `show storageArray autoConfiguration` command returns configuration information in the form of a list of valid physical disk types, RAID levels, virtual disk information, and hot spare information. (This list corresponds to the parameters for the `autoConfigure storageArray` command.)

The RAID controller modules audit the storage array and then determine the highest RAID level that the storage array can support and the most efficient virtual disk definition for the RAID level. If the configuration described by the returned list is acceptable, enter the `autoConfigure storageArray` command without any parameters. To modify the configuration, change a single parameter or all of the parameters to meet your configuration requirements. After entering the `autoConfigure storageArray` command, the RAID controller modules set up the storage array using either the default parameters or those you selected.

Syntax

```
autoConfigure storageArray [physicalDiskType=(SAS)] [raidLevel=(0 | 1 | 5 | 6)]
[diskGroupWidth=numberOfPhysicalDisks] [diskGroupCount=numberOfDiskGroups]
[virtualDisksPerGroupCount=numberOfVirtualDisksPerGroup]
[hotSpareCount=numberOfHotSpares] [segmentSize=segmentSizeValue]
[cacheReadPrefetch=(TRUE | FALSE)] [readAheadMultiplier=multiplierValue]
[securityType=(none | capable | enabled)] [dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`physicalDiskType`	Type of physical disks to use for the storage array. Valid physical disk type is Serial Attached SCSI (SAS). This parameter is not required if only one type of physical disk is in the storage array.
`raidLevel`	RAID level of the disk group that contains the physical disks in the storage array. Valid RAID levels are `0`, `1`, `5` or `6`.
`diskGroupWidth`	Number of physical disks in a disk group in the storage array. For information about the number of physical disks that you can use in a disk group, see Enclosure Loss Protection.
`diskGroupCount`	Number of disk groups in the storage array. Use integer values.

Parameter	Description
`virtualDisksPerGroupCount`	Number of equal-capacity virtual disks per disk group. Use integer values.
`hotSpareCount`	Number of hot spares in the storage array. Use integer values. For information about hot spares, see Assigning Global Hot Spares.
`segmentSize`	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are `8`, `16`, `32`, `64`, `128`, `256`, or `512`. For information about the `segmentSize` parameter, see Using the Auto Configure Command.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to `FALSE`. To turn on cache read prefetch, set this parameter to `TRUE`.
`securityType`	The setting to specify the security level when creating the disk groups and all associated virtual disks. • `none`—The disk group and virtual disks are not secure. • `capable`—The disk group and virtual disks are capable of having security set, but security has not been enabled. • `enabled`—The disk group and virtual disks have security enabled.
`dataAssurance`	The setting to specify that a disk group, and the virtual disks within the disk group, has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk group. These settings are valid: • **none** - No explicit data assurance is defined. The disk group will be comprised of virtual disks that may or may not have data assurance, depending on the availability of data assurance enabled physical disks. Data Assurance is enabled on virtual disks created on disk groups that support data assurance. If only data assurance physical disks are available, the disk groups will be data assurance enabled. • **enabled** - The disk group has data assurance protection. The disk group supports protected information and is formatted with protection information enabled.

Autoconfigure Storage Array Hot Spares

Description

This command automatically defines and configures the hot spares in a storage array. You can run this command at any time. This command provides the best hot spare coverage for a storage array.

Syntax

```
autoConfigure storageArray hotSpares
```

Parameters

None.

NOTE: When you run the `autoconfigure storageArray hotSpares` command, the RAID controller module firmware determines the number of hot spares to create based on the total number and type of physical disks in the storage array.

Blink SSD Cache

Description

The `start locate` command identifies the Solid State Disks (SSDs) that are being used in the SSD cache by turning on the indicator lights for the SSDs. The `stop locate` command turns off the indicator lights on the SSDs.

Syntax Applicable To Starting A Blink Operation

```
start ssdCache [ssdCacheName] locate
```

Syntax Applicable To Stopping A Blink Operation

```
stop ssdCache locate
```

Parameter

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache that you want to locate. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Cancel Remote Replication Group Role Reversal

Description

This command cancels a pending role reversal operation between remote replication groups.

Syntax

```
stop asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"] rolechange
```

Parameter

Parameter	Description
asyncRemoteReplicationGroup	The name of the remote replication group for which you want to cancel the pending role reversal operation. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ").

Change SSD Cache Application Type

Description

This command changes the application type associated with the SSD cache. The application type can be web server, database, or multimedia. Changing the application type changes the block size, sub block size, populate on read threshold, and populate on write threshold for the virtual disks underlying the SSD cache.

Syntax

```
set ssdCache [ssdCacheName] usageHint=(webServer|dataBase|fileSystem)
```

Parameters

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache for which you want to change the application type. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
usageHint	The values are based on the typical I/O usage pattern of the application that is using the SSD

Parameter	Description
	cache. Valid values are `webServer`, `dataBase`, or `fileSystem`.

Check Remote Replication Status

NOTE: This command valid only with Fibre Channel storage arrays.

Description

This command returns the status of a remote replication virtual disk. Use this command to determine when the status of the remote replication virtual disk becomes Optimal.

Syntax

```
check remoteReplication localVirtualDisk [virtualDiskName]
optimalStatus timeout=timeoutValue
```

Parameters

Parameter	Description
`localVirtualDisk`	The name of any remote replication virtual disk. The remote replication virtual disk can be the primary virtual disk or the secondary virtual disk of a remote replication pair. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").
`timeout`	The time interval within which the software can return the remote replication virtual disk status. The timeout value is in minutes.

NOTE: This command waits until the status becomes Optimal or the timeout interval expires.

Check Remote Replication Group Consistency

Description

This command applies to a remote replication group that has underlying repository virtual disks. The command produces a report based on analysis of the data in the repository.

Syntax

```
check asyncRemoteReplicationGroup[asyncRemoteReplicationGroupName]
repositoryConsistency localVirtualDisk=["localVirtualDiskName"]
file="filePath"
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The alphanumeric identifier (including - and _) of the remote replication group on which to run a consistency check. Enclose the remote replication group identifier in square brackets ([ ]).
`localVirtualDisk`	Specifies the alphanumeric identifier of a local virtual disk that participates in the replication relationship. You can specify a local virtual disk that is either the source or the destination on a replication relationship. Enclose the local virtual disk identifier in double quotation marks (" ") within square brackets ([ ]).
`file`	The file path and the file name to which you want to save the report that results from the consistency check. Enclose the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI \logs \repoConsistency.txt"** This parameter must appear last, after any of the optional parameters.

Check Repository Consistency

Description

This command applies to a number of objects that have underlying repository virtual disks. The command produces a report based on analysis of the data in the repository.

Syntax

```
check (snapGroup[snapGroupName] repositoryConsistency | snapVirtualDisk
[snapVirtualDiskName] repositoryConsistency | virtualDisk [virtualDiskName]
repositoryConsistency | virtualDiskCopy target [targetName]
repositoryConsistency | localVirtualDisk="localVirtualDiskName") file="filePath"
```

Parameters

Parameter	Description
`repositoryConsistency`	Specifies that the snapshot group, snapshot virtual disk, or virtual disk (depending on which is specified by the corresponding parameter) is checked for consistency.
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group on which to run a consistency check. Enclose the snapshot group identifier in square brackets ([ ]).

Parameter	Description
`snapVirtualDisk`	The alphanumeric identifier (including - and _) of the snapshot virtual disk on which to run a consistency check. Enclose the snapshot virtual disk identifier in square brackets ([ ]).
`virtualDisk`	The alphanumeric identifier (including - and _) of the thin virtual disk on which to run a consistency check. Enclose the virtual disk identifier in square brackets ([ ]).
`virtualDiskCopy`	Specifies that the target of a virtual disk copy relationship is checked for consistency.
`target`	Use only in conjunction with the `virtualDiskCopy` parameter to specify the alphanumeric identifier of the virtual disk that is the target of the virtual disk copy relationship. Enclose the virtual disk identifier in double quotation marks (" ").
`file`	The file path and the file name to which you want to save the report that results from the consistency check. Enclose the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI \logs \repoConsistency.txt"**. This parameter must appear last, after any of the optional parameters.

Check Storage Array Connectivity

Description

This command verifies that the local storage array and the remote storage array have a communication path and displays the connection details between the local and remote storage array.

Before creating a remote replication group, you should check whether the local storage array and the remote storage array can communicate with each other. When you execute this command, the system queries for the remote storage array on all eligible host ports to determine what ports are connected to the remote storage array. The result of the test is a list of all ports on the local storage array along with a list of the remote storage array port addresses accessible through that port.

NOTE: Connectivity is tested using all possible channels, and if it is a dual RAID controller module configuration, connectivity is checked from each RAID controller module. It might take up to 20 minutes to check connectivity between two storage arrays.

Syntax

```
check storageArray connectivity  (remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwid=<wwID>)
```

Parameters

Parameter	Description
`remoteStorageArrayName`	The name for the remote storage array for which you are checking connectivity. Enclose the storage array name in double quotation marks (" ").
`remoteStorageArrayNameWwid`	The World Wide Identifier (WWID) of the storage array for which you are checking connectivity. You can use the WWID instead of the storage array name to identify the storage array. Enclose the WWID in angle brackets (< >).

Notes

Before a replication can be established between two storage arrays, they must be connected through a Fibre Channel connection or an iSCSI connection.

Fibre Channel remote replication must be activated on both storage arrays before they can communicate with each other for replicating or connectivity checking.

If the local storage array supports iSCSI, connectivity over iSCSI is checked. If the local storage array supports Fibre Channel and replicating over Fibre Channel has been activated, connectivity over Fibre Channel is checked. If the local storage array does not support iSCSI or Fibre Channel, an error message is displayed and the operation aborted.

Check Virtual Disk Consistency

Description

This command checks a virtual disk for consistency and media errors, and writes the results of the check to a file.

Syntax

```
check virtualDisk [virtualDiskName]
consistency [consistencyErrorFile=filename]
[mediaErrorFile=filename]
[priority=(highest | high | medium | low | lowest)]
[startingLBA=LBAvalue] [endingLBA=LBAvalue]
[verbose=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`virtualDisk`	Name of the specific virtual disk to check consistency. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has

Parameter	Description
	special characters, you must also put quotation marks (" ") around the virtual disk name.
consistencyErrorFile	Name of the file in which to save the consistency error information. You must put quotation marks (" ") around the file name.
mediaErrorFile	Name of the file in which to save the media error information. You must put quotation marks (" ") around the file name.
priority	Specifies the priority that the consistency check has relative to host I/O activity. Valid entries are highest, high, medium, low, or lowest.
startingLBA	The starting logical block address. Use integer values.
endingLBA	The ending logical block address. Use integer values.
verbose	Captures progress details, such as percent complete, and shows the information while virtual disk consistency is being repaired. To capture progress details, set this parameter to TRUE. To prevent capturing progress details, set this parameter to FALSE.

NOTE: The starting logical block address and the ending logical block address are useful for very large single-virtual disk LUNs. Running a virtual disk consistency check on a very large single-virtual disk LUN can take a long time. By defining the beginning address and ending address of the data blocks, you can reduce the time that a virtual disk consistency check takes to complete.

Clear Physical Disk Port Statistics

Description

This command resets the statistics for all physical disk ports.

Syntax

```
clear allPhysicalDiskChannels stats
```

Parameters

None.

Clear Remote Replication Fault

Description

This command clears a remote replication "sticky" fault from one or more remote replication groups and one or more remote replication group member virtual disks.

A remote replication group and its member virtual disks can encounter remote replication "sticky" faults, which occur at a single point-in-time but do not impact the functionality of the replications. These type of faults must be reviewed, but do not require any changes to the configuration.

A remote replication group and its member virtual disks can have more than one associated “sticky” fault. This command clears all of the faults associated with the remote replication group and its member virtual disk. However, if a remote replication group has a fault and one of its member virtual disks has a fault, clearing the fault on the remote replication group does not clear the fault on its member virtual disk.

Syntax

```
clear asyncRemoteReplicationFault(all | asyncRemoteReplicationGroup ["
asyncRemoteReplicationGroupName"] |
asyncRemoteReplicationGroups ["asyncRemoteReplicationGroupName1" ...
"asyncRemoteReplicationGroupNameN"] |
virtualDisk ["virtualDiskName"] | virtualDisks ["virtualDiskName1" ...
"virtualDiskNameN"])
```

Parameters

Parameter	Description
`all`	Use this parameter if you want to clear all remote replication faults from all remote replication groups and on all remote replication group member virtual disks.
`asyncRemoteReplicationGroup`	The name of the remote replication group from which you want to clear the remote replication fault. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ").
`asyncRemoteReplicationGroups`	The names of several remote replication groups from which you want to clear the remote replication fault. Enter the names of the remote replication groups using these rules: • Enclose all of the names in square brackets ([ ]). • Enclose each of the names in double quotation marks (" "). • Separate each of the names with a space.

Parameter	Description
`virtualDisk`	The name of the specific member virtual disk (in a remote replication group) from which you want to clear the remote replication fault. Enclose the member virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ") inside square brackets. If the virtual disk name consists only of numbers, such as "1002," you also must enclose the virtual disk name in double quotation marks (" ") inside square brackets.
`virtualDisks`	The names of several member virtual disks (in a remote replication group) from which you want to clear the remote replication fault. Enter the names of the member virtual disks using these rules: • Enclose all of the names in square brackets ([ ]). • Enclose each of the names in double quotation marks (" "). • Separate each of the names with a space.

Clear Storage Array Configuration

Description

Use this command to:

- Clear the entire storage array configuration, and return it back to the initial installation state.
- Clear the configuration except for security information and identification information.
- Clear disk group configuration information and virtual disk configuration information only.

CAUTION: As soon as you run this command, the existing storage array is deleted.

Syntax

```
clear storageArray configuration (all | diskGroups)
```

Parameters

If you do not enter a parameter, this command removes all configuration information for the storage array, except for the information related to security and identification.

Parameter	Description
`all`	The setting to remove the entire configuration of the storage array, including security and identity information. Removing all configuration

Parameter	Description
	information returns the storage array to its initial state.
`diskGroups`	The setting to remove the virtual disk configuration and the disk group configuration. The rest of the configuration stays intact.

NOTE: When you run this command, the storage array becomes unresponsive, and all script processing is cancelled. You must remove and re-add the storage array to resume communication with the host. To remove an unresponsive storage array, access the Enterprise Management Window, and click **Edit → Remove.** To re-add the storage array, access the Enterprise Management Window, click **Edit → Add Storage Array**, and enter the appropriate IP addresses.

Set Storage Array Core Dump

Description

This command sets a flag on a RAID controller module to allow a new core dump to overwrite an existing core dump.

Syntax

```
set storageArray coreDumpAllowOverWrite
```

Parameters

None

Notes

When you retrieve a core dump from the RAID controller module cache to a host, a flag is set on the RAID controller module to indicate that the core dump does not need to be retrieved. This setting persists for 48 hours. If a new core dump occurs during that period, the new core dump is saved to the RAID controller module cache and overwrites any previous core dump data in the cache.

You can use the `set storageArray coreDumpAllowOverWrite` command to set the RAID controller module flag so that a new core dump overwrites any previous core dump. Without retrieving a core dump, this command sets the flag by default.

Clear Storage Array Event Log

Description

This command clears the Major Event Log (MEL) for the storage array by deleting the data in the MEL buffer.

CAUTION: As soon as you run this command, the existing MEL in the storage array is deleted.

Syntax

```
clear storageArray eventLog
```

Parameters

None.

Clear Storage Array Firmware Pending Area

Description

This command deletes a previously downloaded firmware image or nonvolatile static random access memory (NVSRAM) values from the pending area buffer.

CAUTION: As soon as you run this command, the contents of the existing pending area in the storage array are deleted.

Syntax

```
clear storageArray firmwarePendingArea
```

Parameters

None.

Clear Storage Array Recovery Mode

Description

This command forces a storage array to exit recovery mode.

Syntax

```
clear storageArray recoveryMode
```

Notes

Recovery mode is entered during start-of-day operations when the system configuration is cleared and a valid on-board backup is available. This mode is exited by either restoring the system configuration from a backup location, or by clearing the existing on-board backups. While recovery mode is in force, a **needs attention** condition is raised and the **Recovery Guru** becomes available from the user interface. However, the system configuration is empty during recovery mode.

Clear Virtual Disk Reservations

Description

This command clears persistent virtual disk reservations.

Syntax

```
clear (allVirtualDisks | virtualDisk [virtualDiskName]
| virtualDisks [virtualDiskName1... virtualDiskNameN])
reservations
```

Parameters

Parameter	Description
`allVirtualDisks`	Clears reservations on all virtual disks in the storage array.
`virtualDisk` or `virtualDisks`	Name of the specific virtual disk for which to clear reservations. You can enter more than one virtual disk name. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has special characters, you must also put quotation marks (" ") around the virtual disk name.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Clear Virtual Disk Unreadable Sectors

Description

This command clears the unreadable sector information from one or more virtual disks.

Syntax

```
clear (allVirtualDisks | virtualDisk
[virtualDiskName] | virtualDisks [virtualDiskName1
... virtualDiskNameN]) unreadableSectors
```

Parameters

Parameter	Description
`allVirtualDisks`	Clears unreadable sector information from all of the virtual disks in the storage array.
`virtualDisk` or `virtualDisks`	The name of the specific virtual disk for which you want to clear unreadable sector information. You can enter more than one virtual disk name. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Configure Automatic Support Bundle Collection

Description

This command enables or disables the automatic collection of support bundles on one or more storage arrays when a critical Major Event Log (MEL) event is detected.

NOTE: This command must be run from the command line. It cannot be run in the MD Storage Manager script editor.

Syntax

```
supportBundle auto (enable|disable) (all | storageArrayName)
[data=directorypath]
```

Parameters

Parameter	Description
`enable`	Enable automatic collection of support bundles when a critical MEL event is detected.
`disable`	Disable automatic collection of support bundles where the feature was previously enabled.
`all`	Apply the command to all storage arrays detected from the host.
`storageArrayName`	Apply the command to the named storage array.
`data`	Store the data generated by the command at the location specified by the *directorypath*.

Convert Snapshot (Legacy) Virtual Disk To Snapshot Group

Description

This command migrates a snapshot (legacy) virtual disk to a snapshot group and converts an existing snapshot (legacy) repository virtual disk to a snapshot image repository virtual disk. A maximum of four snapshots (legacy) can be converted to a snapshot group. None of the snapshots (legacy) can be in a disabled state or involved in an in-progress virtual disk copy operation. If the base virtual disk has more than four snapshots (legacy), you must delete the extra snapshots before running this command.

Syntax

```
convert snapshotVirtualDisk baseVirtualDisk="baseVirtualDiskName"
```

Parameter

Parameter	Description
`baseVirtualDisk`	The alphanumeric identifier (including - and _) of the base virtual disk that has the snapshots (legacy) that you want to convert into a snapshot group. Enclose the base virtual disk name in double quotation marks (" ").

Create Consistency Group

NOTE: This command does not apply to the snapshot (legacy) commands.

Description

This command creates a new, empty consistency group that can contain snapshot groups. You must add the snapshot groups using the `set consistencyGroup addCGMember` command.

Syntax

```
create consistencyGroup userLabel="consistencyGroupName"
[repositoryFullPolicy=(failBaseWrites repositoryFullLimit=percentValue |
autoDeleteLimit=numberOfSnapImages |
enableSchedule=(TRUE | FALSE) |
schedule (immediate | snapSchedule) |
rollbackPriority=(lowest | low | medium | high | highest)]
```

Parameters

Parameter	Description
`userLabel`	The name of the new consistency group that you want to create. Enclose the new consistency group name in double quotation marks (" ").
`repositoryFullPolicy`	How you want snapshot processing to continue if the snapshot repository virtual disks are full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or delete (purge) the snapshot images (`purgeSnapImages`). The default action is `purgeSnapImages`.
`repositoryFullLimit`	The percentage of repository capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.
`autoDeleteLimit`	The maximum number of snapshot images that you want to automatically delete if you have

Parameter	Description
	selected to purge the snapshot images for a repository full policy. The default value is `32`.
`enableSchedule`	Whether the ability to schedule a snapshot operation is turned on or turned off. To turn on snapshot scheduling, set this parameter to `TRUE`. To turn off snapshot scheduling, set this parameter to `FALSE`.
`schedule`	Use this parameter to schedule a snapshot operation. You can use one of these options for setting a schedule for a snapshot operation: • `immediate` • `startDate` • `scheduleDay` • `startTime` • `scheduleInterval` • `endDate` • `timesPerDay` • `timeZone` See the Notes section for information explaining how to use these options.
`rollBackPriority`	Determines whether system resources should be allocated to the rollback operation at the expense of system performance. A value of high indicates that the rollback operation is prioritized over all other host I/O. A value of `low` indicates that the rollback operation should be performed with minimal impact to host I/O.

Notes

A consistency group is a logical entity that enables you to manage in batch form all of the snapshot images that you add to the collection. The consistency group is a collection of snapshot groups that have mutual consistency requirements or dependencies for their snapshot images. Any snapshot images that you create and use for this collection must be managed in accordance with the consistency dependencies.

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

The snapshot images in a consistency group can be deduced based on the existence of a snapshot image within a consistency group. All snapshot images that reside within a consistency group share a common time stamp and sequence number.

An operation on a snapshot image consistency group is treated as a single request, and causes all pending I/O operations to the associated base virtual disk of each member to be drained and suspended before creating the snapshot images. If creation of the snapshot images cannot be completed successfully for all of the consistency group members, the operation fails and has no affect (that is, new snapshot images are not created).

Based on this behavior all members for a consistency group usually have the same number of PiTs. However, when a new member is added to a consistency group, that new member lacks the snapshot images that were previously created on the established members of the consistency group. This is not considered an error condition. Ensuing requests for deletion or rollback of snapshot images that only exist on a subset of the consistency group members will only affect the members for which the specified snapshot images actually exists.

For more information on scheduling snapshot images in a consistency group, see Scheduling Snapshots.

Create Consistency Group Snapshot Image

Description

This command creates a new snapshot image for each base virtual disk that is a member of a snapshot consistency group.

Syntax

```
create cgSnapImage consistencyGroup="consistencyGroupName"
```

Parameter

Parameter	Description
`consistencyGroup`	The name of the consistency group for which you are creating snapshot images. Enclose the consistency group name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

The command causes all pending I/O operations to each base virtual disk that is a member of the consistency group to be drained and suspended before creating the snapshot images. If the creation of all of the snapshot images cannot be completed successfully for all of the consistency group members, the operation fails and new snapshot images are not created.

Normally, all members of a snapshot consistency group will have the same number of snapshot images. When you add a new member to a snapshot consistency group, that new member lacks the snapshot images that were previously created on the established members of the snapshot consistency group. This is not an error condition. Requests for deletion or rollback of snapshot images that exist on only a subset of the snapshot consistency group members affects only those members for which the specified snapshot image actually exists.

Create Consistency Group Snapshot Virtual Disk

Description

This command creates a snapshot virtual disk of specific images in the base virtual disks in a consistency group. You can select one base virtual disk or more than one base virtual disks from the consistency group to include in the snapshot virtual disk. When you create a snapshot virtual disk of a consistency group, you are creating a virtual disk with contents that you can view.

Syntax With User Specified Base Virtual Disks

```
create cgSnapVirtualDisk userLabel="cgVirtualDiskName"
cgSnapImageID="snapCGID:imageID"
members=(baseVirtualDisk:repositoryVirtualDisk...baseVirtualDisk:repositoryVirtu
alDisk)
```

Syntax When Setting The Consistency Group Snapshot Virtual Disk To Read Only

```
create cgSnapVirtualDisk userLabel="cgVirtualDiskName"
cgSnapImageID="snapCGID:imageID"
readOnly
```

Syntax When Setting The Repository Full Limit

```
create cgSnapVirtualDisk userLabel="cgVirtualDiskName"
cgSnapImageID="snapCGID:imageID"
repositoryFullLimit=percentValue
```

Parameters

Parameter	Description
`userLabel`	The name that you want to give the consistency group snapshot virtual disk that you are creating. Enclose the consistency group snapshot virtual disk name in double quotation marks (" ").
`cgSnapImageID`	The identifier of the snapshot image from the snapshot consistency group that you want to include in the consistency group snapshot virtual disk. Enclose the snapshotImage identifier in double quotation marks (" ").
`members`	The identifier of one base virtual disk or more than one base virtual disks that you want to add. The members identifier is comprised of the base virtual disk name concatenated with the repository virtual disk name. You must use a colon (:) between the two names. Enclose all of the member identifiers in parentheses. If you enter more than one member separated the members with a space.

Parameter	Description
	When you do not use the members parameter, all of the members are automatically added to the new consistency group snapshot virtual disk.
`repositoryFullLimit`	The percentage of repository capacity at which the consistency group snapshot repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent.
`readOnly`	The setting to determine whether you can write to the snapshot virtual disk or only read from the snapshot virtual disk. To write to the snapshot virtual disk, do not include this parameter. To prevent writing to the snapshot virtual disk, include this parameter.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

If you do not specify the `repositoryVolType` or `readOnly` parameters, the MD storage management software selects the repositories for the consistency group snapshot virtual disk. If the disk group or disk pool where the base virtual disk resides does not have enough space, this command fails.

The `create cgSnapVirtualDisk` command has unique forms that are explained by these examples:

- Creating a read/write consistency group snapshot virtual disk on a snapshot consistency group named **snapCG1** that has three members cgm1, cgm2, and cgm3. The repository virtual disks already exist and selected by the user in this command.

  ```
  create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk1"
  cgSnapImageID="snapCG1:oldest" members=(cgm1:repos_0010 cgm2:repos_0011
  cgm3:repos_0007);
  ```

  Note the use of the colon (:) in the name of the snapshot image to be included in the consistency group snapshot virtual disk. The colon is a delimiter that separates the name of the snapshot virtual disk from a particular snapshot image that you might want to use. You can use one of these options following the colon:

  - An integer value that is the actual sequence number of the snapshot image.
  - newest - Use this option when you want to show the latest consistency group snapshot image.
  - oldest - Use this option when you want to show the earliest snapshot image created.

  The use of the colon following the names of the members of the snapshot consistency group acts define the mapping between the member and a repository virtual disk. For example, in cgm1:repos1_10, member cgm1 maps to repository virtual disk repos1_0010.
- Creating a read/write consistency group snapshot virtual disk on a snapshot consistency group named **snapCG1** of only members cgm1 and cgm2:

  ```
  create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk2"
  cgSnapImageID="snapCG1:14214" members=(cgm1:repos_1000 cgm2:repos_1001);
  ```

- Creating a read-only consistency group snapshot virtual disk on a snapshot consistency group named **snapCG1** that has three memebers cgm1, cgm2, and cgm3:

  ```
  create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk3"
  cgSnapImageID="snapCG1:oldest" readOnly;
  ```

- Creating a consistency group snapshot virtual disk that has a repository full limit set to 60 percent on a snapshot consistency group named **snapCG1** that has three members cgm1, cgm2, and cgm3:

  ```
  create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk3"
  cgSnapImageID="snapCG1:oldest" repositoryFullLimit=60;
  ```

- Creating a read/write consistency group snapshot virtual disk with automatic repository selection on a snapshot consistency group named **snapCG1** that has three memebers cgm1, cgm2, and cgm3:

  ```
  create cgSnapVirtualDisk userLabel="cgSnapVirtualDisk4"
  cgSnapImageID="snapCG1:oldest";
  ```

Create Consistency Group Snapshot Virtual Disk Mapping

Description

This command creates a logical mapping from a consistency group snapshot virtual disk to a host or a host group.

Syntax

```
create mapping cgSnapVirtualDisk="snapVirtualDiskName" (host="hostName" |
hostGroup=("hostGroupName" | defaultGroup))
```

Parameters

Parameter	Description
`cgSnapVirtualDisk`	The name of the consistency group snapshot virtual disk for which you want to create a logical mapping. Enclose the consistency group snapshot virtual disk name in double quotation marks (" ").
`host`	The name of a host to which you want to create a logical mapping. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of a host group to which you want to create a logical mapping. Enclose the host group name in double quotation marks (" "). If you use the `defaultGroup` keyword, do not enclose it in quotation marks.

Notes

A host is a computer that is attached to the storage array and accesses the virtual disks on the storage array through the host ports. You can define specific mappings to an individual host. You also can assign the host to a host group that shares access to one or more virtual disks.

A host group is an optional topological element that you can define if you want to designate a collection of hosts that share access to the same virtual disks. The host group is a logical entity. Define a host group only if you have two or more hosts that share access to the same virtual disks.

Create Disk Group

Description

This command creates either a free-capacity disk group or a disk group with one virtual disk when you enter a set of assigned physical disks.

Syntax

```
create diskGroup
physicalDisks=(enclosureID1,drawerID1,slotID1
enclosureID2,slotID2... enclosureIDn,drawerIDn,slotIDn)
raidLevel=(0 | 1 | 5 | 6)
userLabel="diskGroupName"
[driveMediaType=(HDD | SSD | unknown | allMedia)
capacity=virtualDiskCapacity
owner=(0 | 1)
cacheReadPrefetch=(TRUE | FALSE)
segmentSize=segmentSizeValue
usageHint=(fileSystem | dataBase | multiMedia)
enclosureLossProtect=(TRUE | FALSE)
drawerLossProtect=(TRUE | FALSE)
securityType=(none | capable | enabled)]
[dataAssurance=(none | capable | enabled)]
```

Parameters

Parameter	Description
`physicalDisks`	The physical disks that you want to assign to the disk group that you want to create. Specify the enclosure ID value and the slot ID value for each physical disk that you assign to the disk group. Enclosure ID values are `0` to `99`. Drawer ID values are `0` to `4`. Slot ID values are `0` to `31`. Enclose the enclosure ID values and the slot ID values in parentheses.
`raidLevel`	The RAID level of the disk group that contains the virtual disk. Valid values are `0`, `1`, `5`, or `6`.
`userLabel`	The alphanumeric identifier (including - and _) that you want to give the new disk group. Enclose the disk group identifier in double quotation marks (" ").
`driveMediaType`	The type of physical disk media that you want to use for the disk group. Valid physical disk media consists of: • HDD — Use this option when you have hard disk drives in the expansion enclosure. • SSD — Use this option when you have solid state physical disks in the expansion enclosure. • unknown —Use if you are not sure what types of physical disk media are in the expansion enclosure.

Parameter	Description
	• allMedia — Use this option when you want to use all types of physical disk media that are in the expansion enclosure.
physicalDiskType	The type of physical disk that you want to use in the virtual disk. You cannot mix physical disk types. The valid physical disk type is **SAS**. You must specify a physical disk type.

capacity	The size of the virtual disk that you are adding to the storage array. Size is defined in units of bytes, KB, MB, GB, or TB. The following examples show the syntax:
	`capacity=500 bytes`
	`capacity=2 GB`
	NOTE: A space must be added between the last digit and the size (bytes, KB, MB, or GB) for values greater than 9. If you do not specify a capacity, all the available physical disk capacity in the disk group is used. If you do not specify the capacity unit, bytes is used as the default unit.

owner	The setting for which RAID controller module owns the disk group. Valid RAID controller module identifier values are 0 and 1. The identifier value is 0 for the RAID controller module on the top and 1 for the RAID controller module on the bottom when viewed from the rear of the enclosure. If you do not specify an owner, the RAID controller module firmware determines the owner.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
segmentSize	The amount of data (in KB) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512.
usageHint	The setting for both the cacheReadPrefetch parameter and the segmentSize parameter to be default values. The default values are based on the typical I/O usage pattern of the application that is using the virtual disk. Valid values are fileSystem, dataBase, or multiMedia.
enclosureLossProtect	The setting to enforce enclosure loss protection when you create the disk group. To enforce enclosure loss protection, set this parameter to TRUE. The default value is FALSE.
drawerLossProtect	The setting to enforce drawer loss protection when you create the disk group. To enforce drawer loss protection, set this parameter to TRUE. The default value is FALSE.
securityType	The setting to specify the security level when creating the disk groups and all associated virtual disks. • none—The disk group and virtual disks are not secure.

Parameter	Description
	• `capable`—The disk group and virtual disks are capable of having security set, but security has not been enabled. • `enabled`—The disk group and virtual disks have security enabled.
`dataAssurance`	The setting to specify that a disk group, and the virtual disks within the disk group, has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk group. These settings are valid: • **none** - The disk group does not have data assurance protection. • **enabled** - The disk group has data assurance protection. The disk group supports protected information and is formatted with protection information enabled.

Additional Information

Cache Read Prefetch

The `cacheReadPrefetch` command lets the RAID controller module copy additional data blocks into cache while the RAID controller module reads and copies data blocks that are requested by the host from the physical disks into cache. This action increases the chance that a future request for data can be fulfilled from cache. `CacheReadPrefetch` is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the RAID controller module reads into cache. Valid values for the `cacheReadPrefetch` parameter are `TRUE` or `FALSE`.

You do not need to enter a value for the `cacheReadPrefetch` parameter or the `segmentSize` parameter. If you do not enter a value, the RAID controller module firmware uses the `usageHint` parameter with `fileSystem` as the default value. Entering a value for the `usageHint` parameter and a value for the `cacheReadPrefetch` parameter or a value for the `segmentSize` parameter does not cause an error. The value that you enter for the `cacheReadPrefetch` parameter or the `segmentSize` parameter takes priority over the value for the `usageHint` parameter.

Segment Size

The size of a segment determines how many data blocks that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8 KB segment holds 16 data blocks. A 64 KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the RAID controller module at run time. If the value that you entered is not valid, the RAID controller module returns a list of valid values. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests.

If the virtual disk is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data

stripe. A data stripe is the segment size that is multiplied by the number of physical disks in the disk group that are used for data transfers. In this case, multiple physical disks are used for the same request, but each physical disk is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of physical disks that are required to satisfy a data transfer request.

Security Type

The `securityType` parameter is valid for physical disks that are capable of selfencrypting disk (SED). With SED, the RAID controller module firmware can create a key and activate the physical disk security feature. The physical disk security feature encrypts data as the data is written to the physical disk and decrypts the data as the data is read from the physical disk. Without the key created by the RAID controller module, the data written to the physical disk is inaccessible.

Before you can set the `securityType` parameter to capable or enabled, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. The following commands are related to the security key:

- `create storageArray securityKey`
- `set storageArray securityKey`
- `start secureErase (physicalDisk | physicalDisks)`
- `enable diskGroup [diskGroupName] security`

Enclosure Loss Protection and Drawer Loss Protection

For enclosure loss protection to work, each physical disk in a disk group must be in a separate enclosure. If you set the `enclosureLossProtect` parameter to `TRUE` and have selected more than one physical disk from any one enclosure, the storage array returns an error. If you set the `enclosureLossProtect` parameter to `FALSE`, the storage array performs operations, but the disk group that you create might not have enclosure loss protection. Enclosure loss protection is not valid when you create virtual disks on existing disk groups.

The `drawerLossProtect` parameter defines if data on a virtual disk is accessible if a drawer fails. When you assign the physical disks, if you set the `drawerLossProtect` parameter to `TRUE` and select more than one physical disk from any one drawer, the storage array returns an error. If you set the `drawerLossProtect` parameter to `FALSE`, the storage array performs operations, but the disk group that you create might not have drawer loss protection. You must set the `enclosureLossProtect` parameter and the `drawerLossProtect` parameter to the same value. Either both the parameters must be `TRUE` or both must be `FALSE`. If the `enclosureLossProtect` parameter and the `drawerLossProtect` parameter are set to different values, the storage array returns an error.

Create Disk Pool

Description

This command creates a new disk pool based on the specified parameters. You can create the disk pool by entering either a list of physical disks or a type of physical disk that you want to use for the disk pool. Each disk pool must be created from a minimum of 11 physical disks specified. There is no upward limit of physical disks supported in a single disk pool, but the number cannot exceed the number of physical

disks supported on the storage array or the maximum size allowed by the model. iSCSI units have a limit of 256 TB and SAS and Fibre Channel units have a limit of 1024 TB.

NOTE: If you enter a list of physical disks, ensure that all of the physical disks have the same capacity. If the physical disks do not have the same capacity, each physical disk in the disk pool reports capacity equal to the smallest physical disk.

Syntax

Parameters

```
create diskPool
physicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
physicalDiskType=SAS
userLabel="diskPoolName" [physicalDiskCount=physicalDiskCountValue]
[enclosureLossProtect=(TRUE | FALSE)]
[drawerLossProtect=(TRUE | FALSE)]
[allocateReservedSpace=(TRUE | FALSE)]
[warningThreshold=(warningThresholdValue|default)]
[criticalThreshold=(criticalThresholdValue|default)]
[criticalPriority=(highest|high|medium|low|lowest)]
[backgroundPriority=(highest|high|medium|low|lowest)]
[degradedPriority=(highest|high|medium|low|lowest)]
[securityType=(none|capable|enabled)]
[physicalDiskMediaType=(hdd | ssd | allMedia | unknown)]
[dataAssurance=(none|enabled)]
```

Parameter	Description
`physicalDisks`	The physical disks you want to assign to the disk pool. Enclosure ID values are `0` to `99`; drawer ID values are `0` to `4`; slot ID values are `0` to `31`. NOTE: A disk pool must contain at least 11 physical disks.
`physicalDiskType`	Physical disk type value is `sas`. Enclose the physical disk type values in parentheses. If the physical disk type is specified, then all of the unassigned physical disks of that type that are compatible with each other are used for the creation of the disk pool.
`physicalDiskCount`	The `physicalDiskCount` parameter limits the disk pool candidates to the given number. Use this parameter only when the `physicalDiskType` parameter is specified.
`reservePhysicalDiskCount`	The default value is automatically determined by the RAID controller module. This value is specified in terms of the number of physical disks to reserve in the disk pool for physical disk failures or physical disk removals.

Parameter	Description
warningThreshold	The percentage of storage capacity at which you receive a warning alert that the disk pool is nearing full capacity. Use integer values. For example, a value of 60 means 60 percent. The default value is 50. Setting this parameter to 0 disables warning alerts. If you set this to default, the warning alert threshold value is determined by the RAID controller module firmware.
criticalPriority	The default value is highest.
backgroundPriority	The default value is low.
securityType	The setting to specify the security level when creating the disk pool. All virtual disk candidates for the disk pool have the specified security type. These settings are valid: • none—The virtual disk candidates are not secure. • capable—The virtual disk candidates are capable of having security set, but security has not been enabled. • enabled—The virtual disk candidates have security enabled. The default value is none.
physicalDiskMediaType	You must use this parameter when you have more than one type of physical disk media in your storage array. Valid physical disk media are: • hdd—Use this option when you have hard disk drives. • ssd—Use this option when you have solid state disks. • unknown—Use this option if you are not sure what types of physical disk media are in the expansion enclosure. • allMedia—Use this option when you want to use all types of physical disk media that are in the expansion enclosure. The default value is hdd.
userLabel	The name that you want to give the new disk pool. Enclose the disk pool name in double quotation marks (" ").
degradedPriority	The priority for degraded activities on the disk pool. For example, disk pool reconstruction after one physical disk failures.

Parameter	Description
	Valid values are **highest**, **high**, **medium**, **low**, and **lowest**. The default value is **high**.
`dataAssurance`	The setting to specify that a disk pool has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk pool. These settings are valid: • **none** – The disk pool does not have data assurance protection. • **enabled** – The disk pool has data assurance protection. The disk pool supports protected information and is formatted with protection information enabled. The default value is none.
`allocateReservedSpace`	The setting to disable or enable the reserve capacity on the physical disk drive in a disk pool. The RAID controller module firmware automatically reserves additional capacity on each physical disk drive when creating a disk pool. To prevent the RAID controller module firmware from reserving additional capacity on each physical disk drive in a disk pool, set this parameter to `FALSE`. The default value is `TRUE`.
`enclosureLossProtect`	The setting to enforce enclosure loss protection when you create the disk pool. To enforce enclosure loss protection, set this parameter to `TRUE`. The default value is `FALSE`.
`drawerLossProtect`	The setting to enforce drawer loss protection when you create the disk pool. To enforce drawer loss protection, set this parameter to `TRUE`. The default value is `FALSE`.

Security Type

The `securityType` parameter is valid for physical disks that are capable of self encrypting disk (SED). With SED, the RAID controller module firmware can create a key and activate the physical disk security feature. The physical disk security feature encrypts data as the data is written to the physical disk and decrypts the data as the data is read from the physical disk. Without the key created by the RAID controller module, the data written to the physical disk is inaccessible.

Before you can set the `securityType` parameter to capable or enabled, you must create a storage array security key. Use the `create storageArray securityKey` command to create a storage array security key. These commands are related to the security key:

- `create storageArray securityKey`
- `enable diskGroup [diskGroupName] security`

- `export storageArray securityKey`
- `import storageArray securityKey`
- `set storageArray securityKey`
- `start secureErase (physicalDisk | physicalDisks)`

Create Host

Description

This command creates a new host. If you do not specify a host group in which to create the new host, the new host is created in the default group.

Syntax

```
create host userLabel="hostName"
[hostGroup=("hostGroupName" | defaultGroup)]
[hostType=(hostTypeIndexLabel |
hostTypeIndexNumber)]
```

Parameters

Parameter	Description
`userLabel`	The name that you want to give the host that you are creating. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group in which you want to create a new host. Enclose the host group name in double quotation marks (" "). (If a host group does not exist, you can create a new host group by using the `create hostGroup` command.) The **defaultGroup** option is the host group that contains the host to which the virtual disk is mapped.
`hostType`	The index label or the index number that identifies the host type. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers. If the host type has special characters, enclose the host type in double quotation marks (" ").

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Create Host Group

Description

This command creates a new host group.

Syntax

```
create hostGroup userLabel="hostGroupName"
```

Parameter

Parameter	Description
`userLabel`	Name to give the host group that you are creating. You must put quotation marks (" ") around the host group name.

Create Host Port

Description

This command creates a new HBA host port identification. The identification is a software value that represents the physical HBA host port to the RAID controller module. Without the host port identification, the RAID controller module cannot receive instructions or data from the host port.

Syntax

```
create hostPort identifier="wwid"
userLabel="portLabel"
host="hostName"
interfaceType=SAS
```

Parameters

Parameter	Description
`identifier`	WWID of the HBA host port. You must put quotation marks (" ") around the WWID.
`userLabel`	Name to give the new HBA host port. You must put quotation marks (" ") around the port label.
`host`	Name of the host for which you are defining an HBA host port. You must put quotation marks (" ") around the host name. **NOTE:** An HBA host port is a physical connection on a host adapter that resides within a host system. An HBA host port provides host access to the virtual disks in a storage array. If the host bus adapter has only one physical connection (one host port), the terms host port and host bus adapter are synonymous.
`interfaceType`	The identifier for the type of interface for the host port. You must enter `SAS`. If you do not enter `SAS`, the command attempts to use an invalid type of interface. A SAS selection requires an 8-byte WWID.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Create iSCSI Initiator

Description

This command creates a new iSCSI initiator object.

Syntax

```
create iscsiInitiator iscsiName = "iSCSI-ID"
userLabel = "name"
host = "host-name"
[chapSecret = securityKey]
```

Parameters

Parameter	Description
`iscsiName`	The default identifier of the iSCSI initiator.
`userLabel`	The name that you want to use for the iSCSI initiator. Enclose the name in double quotation marks (" ").
`host`	The name of the host in which the iSCSI initiator is installed.
`chapSecret`	The password that you want to use to authenticate a peer connection.

NOTE: Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a “secret.” A secret is a security key that is similar to a password. Use `chapSecret` only for initiators requiring mutual authentication.

Create RAID Virtual Disk (Automatic Physical Disk Select)

Description

This command creates a disk group across the storage array physical disks, and a new virtual disk in the disk group. The RAID controller modules in the storage array choose the physical disks to include in the virtual disk.

Syntax

```
create virtualDisk physicalDiskCount=numberOfPhysicalDisks
diskGroupUserLabel="diskGroupName"
raidLevel=(0 | 1 | 5 | 6)
userLabel=”virtualDiskName”
physicalDiskMediaType=(HDD | SSD | unknown | allMedia)
[physicalDiskType=(SAS)]
[capacity=virtualDiskCapacity]
[owner=(0 | 1)]
[cacheReadPrefetch=(TRUE | FALSE)]
[segmentSize=segmentSizeValue]
```

```
[usageHint=(fileSystem | dataBase | multiMedia)]
[enclosureLossProtect=(TRUE | FALSE)]
[drawerLossProtect=(TRUE | FALSE)]
[dssPreAllocate=(TRUE | FALSE)]
[securityType=(none | capable | enabled)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`physicalDiskCount`	Number of unassigned physical disks to use in the disk group. NOTE: The `physicalDiskCount` parameter enables you to choose the number of physical disks to use in the disk group. You do not need to specify the physical disks by enclosure ID and slot ID. The RAID controller modules choose the specific physical disks to use for the disk group.
`diskGroupUserLabel`	The alphanumeric identifier (including – and _) that you want to give the new disk group. Enclose the disk group identifier in double quotes (" ").
`raidLevel`	RAID level of the disk group that contains the virtual disk. Valid values are `0`, `1`, `5` or `6`.
`userLabel`	Name to give to the new virtual disk. You must put quotation marks (" ") around the new virtual disk name.
`physicalDiskType`	Specifies the type of physical disk to use in the virtual disk. You cannot mix physical disk types in the virtual disk. The valid physical disk type is `SAS`.
`capacity`	The size of the virtual disk that you are adding to the storage array. Size is defined in units of bytes, `KB`, `MB`, `GB`, or `TB`. NOTE: A space must be added between the last digit and the size (bytes, KB, MB, or GB) for values greater than 9. If you do not specify a capacity, all the available physical disk capacity in the disk group is used. If you do not specify the capacity unit, bytes is used as the default unit.
`owner`	The RAID controller module that owns the virtual disk. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. If you do not specify an owner, the RAID controller module firmware determines the owner. NOTE: The `owner` parameter defines which RAID controller module owns the virtual disk.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to `FALSE`. To turn on cache read prefetch, set this parameter to `TRUE`.

Parameter	Description
segmentSize	Amount of data (in KB) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512. For information about the segmentSize parameter, see Using The Auto Configure Command.
enclosureLossProtect	Specifies that enclosure loss protection is enforced when creating the disk group. To enforce enclosure loss protection, set this parameter to TRUE. The default setting is FALSE. For information about the enclosureLossProtect parameter, see Enclosure Loss Protection.
drawerLossProtect	The setting to enforce drawer loss protection when you create the replication repository disk group. To enforce drawer loss protection, set this parameter to TRUE. The default value is FALSE.
dssPreAllocate	The setting to make sure that reserve capacity is allocated for future segment size increases. The default value is TRUE.
securityType	The setting to specify the security level when creating the disk groups and all associated virtual disks. • none—The disk group and virtual disks are not secure. • capable—The disk group and virtual disks are capable of having security set, but security has not been enabled. • enabled—The disk group and virtual disks have security enabled. For information about securityType, see Security Type.
usageHint	The setting for both *cacheReadPrefetch* parameter and the *segmentSize* parameter to be default values. The default values are based on the typical I/O usage pattern of the application that is using the virtual disk. Valid values are **fileSystem**, **dataBase**, or **multiMedia**.
dataAssurance	The setting to specify that a disk pool has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk pool. These settings are valid: • **none** – The disk pool does not have data assurance protection. • **enabled** – The disk pool has data assurance protection. The disk pool supports protected information and is formatted with protection information enabled. The default value is none.

Create RAID Virtual Disk (Free Extent Base Select)

Description

This command creates a virtual disk in the free space of a disk group.

Syntax

```
create virtualDisk diskGroup="diskGroupName" userLabel="virtualDiskName"
[freeCapacityArea=freeCapacityIndexNumber] [capacity=virtualDiskCapacity]
[owner=(0 | 1)] [cacheReadPrefetch=(TRUE | FALSE)]
[segmentSize=segmentSizeValue] [usageHint=(fileSystem | dataBase | multiMedia)]
[dssPreAllocate=(TRUE | FALSE)] [securityType=(none | capable | enabled)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`diskGroup`	Sequence number of the disk group in which to create the new virtual disk. (To determine the sequence numbers of the disk groups in your storage array, enter the `show storageArray Profile` command.)
`userLabel`	Name for the new virtual disk. You must put quotation marks (" ") around the new virtual disk name. **NOTE:** You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Spaces are not allowed. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.
`freeCapacityArea`	Index number of the free space in an existing disk group to use to create the new virtual disk. Free capacity is defined as the free capacity between existing virtual disks in a disk group. For example, a disk group might have the following areas: virtual disk 1, free capacity, virtual disk 2, free capacity, virtual disk 3, free capacity. To use the free capacity following virtual disk 2, specify: `freeCapacityArea=2` Use the `show diskGroup` command to determine if the free capacity area exists.

Parameter	Description
`capacity`	Size of the virtual disk that you are adding to the storage array. Size is defined in units of bytes, KB, MB, GB, or TB. NOTE: Add a space between the last digit and the size (bytes, KB, MB, or GB) for values greater than 9. If you do not specify a capacity, all the available physical disk capacity in the disk group is used. If you do not specify the capacity unit, bytes is used as the default unit.
`owner`	RAID controller module that owns the virtual disk. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. If you do not specify an owner, the RAID controller module firmware determines the owner. NOTE: The `owner` parameter defines which RAID controller module owns the virtual disk. The preferred RAID controller module ownership of a virtual disk is the RAID controller module that currently owns the disk group.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn on cache read prefetch, set this parameter to `TRUE`. To turn off cache read prefetch, set this parameter to `FALSE`.
`segmentSize`	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are `8`, `16`, `32`, `64`, `128`, `256`, or `512`. For information about the `segmentSize` parameter, see Using The Auto Configure Command.
`usageHint`	The settings for both the `cacheReadPrefetch` parameter and the `segmentSize` parameter should be the default values. The default values are based on the typical I/O usage pattern of the application that is using the virtual disk. Valid values are `fileSystem`, `dataBase`, or `multiMedia`.
`dssPreAllocate`	The setting to make sure that reserve capacity is allocated for future segment size increases. The default value is `TRUE`.

Parameter	Description
securityType	The setting to specify the security level when creating the disk groups and all associated virtual disks. • none—The disk group and virtual disks are not secure. • capable—The disk group and virtual disks are capable of having security set, but security has not been enabled. • enabled—The disk group and virtual disks have security enabled. For information about securityType, see Security Type.
dataAssurance	The setting to specify that a disk pool has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk pool. These settings are valid: • **none** – The disk pool does not have data assurance protection. • **enabled** – The disk pool has data assurance protection. The disk pool supports protected information and is formatted with protection information enabled. The default value is none.

Create RAID Virtual Disk (Manual Physical Disk Select)

Description

This command creates a new disk group and virtual disk, and enables you to specify the physical disks for the virtual disk.

NOTE: You cannot use mixed physical disk types in the same disk group and virtual disk. This command fails if you specify different types of physical disks for the RAID virtual disk.

Syntax

```
create virtualDisk physicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
diskGroupUserLabel="diskGroupName"
raidLevel=(0 | 1 | 5 | 6)
userLabel="virtualDiskName"
[capacity=virtualDiskCapacity]
[owner=(0 | 1)] [cacheReadPrefetch=(TRUE | FALSE)]
[segmentSize=segmentSizeValue]
[usageHint=(fileSystem | dataBase | multiMedia)]
[enclosureLossProtect=(TRUE | FALSE)]
[drawerLossProtect=(TRUE | FALSE)]
[dssPreAllocate=(TRUE | FALSE)]
```

```
[securityType=(none | capable | enabled)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`physicalDisks`	Specifies the physical disks to assign to the created virtual disk. Specify the enclosure ID and slot ID for each unassigned physical disk to assign to the virtual disk. Enclosure ID values are `0` to `99`. Drawer ID values are `0` to `4`. Slot ID values are `0` to `31`. You must put parentheses around the enclosure ID values and the slot ID values.
`diskGroupUserLabel`	The alphanumeric identifier (including – and _) that you want to give the new disk group. Enclose the disk group identifier in square brackets ([]).
`raidLevel`	RAID level of the disk group that contains the virtual disk. Valid values are `0`, `1`, `5`, or `6`. **NOTE:** If you set the `raidLevel` parameter to RAID 1, the RAID controller module firmware takes the list of physical disks and pairs them using the following algorithm: Data physical disk = *X* Consistency physical disk = *N* / 2 + *X* where, *X* goes from 1 to *N* / 2 and *N* is the number of physical disks in the list. The following data show an example of six physical disks and their replication pairs. **Data** — **Consistency** **1** — *N* / 2 + 1 = 4 **2** — *N* / 2 + 1 = 5 **3** — *N* / 2 + 1 = 6
`userLabel`	Name that you want to give the new virtual disk. You must put quotation marks (" ") around the new virtual disk name. **NOTE:** You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Spaces are not allowed. Names can have a maximum of 30 characters.

Parameter	Description
capacity	Size of the virtual disk that you are adding to the storage array. Size is defined in units of bytes, KB, MB, GB, or TB. NOTE: If you do not specify a capacity, all physical disk capacity available in the disk group is used. If you do not specify capacity units, bytes are used as the default. A space must be added between the last digit and the size (MB, GB, or KB) for values greater than 9.
owner	RAID controller module that owns the virtual disk. Valid RAID controller module identifier values are 0 and 1. The identifier value is 0 for the RAID controller module on the top and 1 for the RAID controller module on the bottom when viewed from the rear of the enclosure. If you do not specify an owner, the RAID controller module firmware determines the owner. NOTE: The owner parameter defines which RAID controller module owns the virtual disk. The preferred owner of a virtual disk is the RAID controller module that currently owns the disk group.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn on cache read prefetch, set this parameter to TRUE. To turn off cache read prefetch, set this parameter to FALSE.
segmentSize	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512. For information about the segmentSize parameter, see Using The Auto Configure Command.
usageHint	The settings for both the cacheReadPrefetch parameter and the segmentSize parameter should be the default values. The default values are based on the typical I/O usage pattern of the application that is using the virtual disk. Valid values are fileSystem, dataBase, or multiMedia.
dssPreAllocate	The setting to make sure that reserve capacity is allocated for future segment size increases. The default value is TRUE.

Parameter	Description
securityType	The setting to specify the security level when creating the disk groups and all associated virtual disks. • none—The disk group and virtual disks are not secure. • capable—The disk group and virtual disks are capable of having security set, but security has not been enabled. • enabled—The disk group and virtual disks have security enabled. For information about securityType, see Security Type.
dataAssurance	The setting to specify that a disk pool has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk pool. These settings are valid: • **none** – The disk pool does not have data assurance protection. • **enabled** – The disk pool has data assurance protection. The disk pool supports protected information and is formatted with protection information enabled. The default value is none.
drawerLossProtect	The setting to enforce drawer loss protection when you create the replicated repository virtual disk. To enforce drawer loss protection, set this parameter to **TRUE**. The default value is **FALSE**.
enclosureLossProtect	The setting to enforce enclosure loss protection when you create the repository. To enforce enclosure loss protection, set this parameter to **TRUE**. The default value is **FALSE**.

Create RAID Virtual Disk On A Disk Pool

Description

This command creates a new RAID virtual disk on an existing disk pool.

Syntax

```
create virtualDisk diskPool="diskPoolName" userLabel="virtual diskName"
capacity=virtualDiskCapacity [thinProvisioned=(TRUE | FALSE)] [owner=(0|1)]
[mapping=(none|default)] [dataAssurance=(none|enabled)]
[(existingRepositoryLabel=existingRepositoryName |
[newRepositoryCapacity=newRepositoryCapacityValue (KB | MB | GB | TB | Bytes))]
```

```
[repositoryMaxCapacity=repositoryMaxCapacityValue (KB|MB|GB|TB|Bytes)]
[warningThresholdPercent=warningThresholdPercentValue]
[repositoryExpansionPolicy=(automatic|manual)] [cacheReadPrefetch=(TRUE |
FALSE)]
```

Parameters

Parameter	Description
`capacity`	The size of the virtual disk that you are creating. Size is defined in units of `MB`, `GB`, or `TB`. The following are examples of the syntax: • `capacity=500MB` • `capacity=2GB`
`thinProvisioned`	This parameter enables thin provisioning for the new virtual disk. To use thin provisioning, set this parameter to `TRUE`. For no thin provisioning, set this parameter to `FALSE`.
`owner`	The RAID controller module that owns the virtual disk. Valid RAID controller module identifiers are `0` or `1`, where `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1. If you do not specify an owner, the RAID controller module firmware determines the owner.
`mapping`	This parameter enables you to map the virtual disk to a host. If you want to map later, set this parameter to `none`. If you want to map the virtual disk now, set this parameter to `default`, and the virtual disk is mapped to all hosts that have access to the storage pool. The default option is `none`.
`warningThreshold`	Valid values are from `1` to `100`. The default value is determined by the RAID controller module firmware. Setting this parameter to `100` disables warning alerts.
`repositoryExpansionPolicy`	Valid values are `automatic` and `manual`.
`cacheReadPrefetch`	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to `FALSE`. To turn on cache read prefetch, set this parameter to `TRUE`.
`segmentSize`	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are `8`, `16`, `32`, `64`, `128`,

Parameter	Description
	256, or 512. For information about the segmentSize parameter, see Using the Auto Configure Command.
usageHint	Valid values are multiMedia, database or fileSystem.

Create Read-Only Snapshot Virtual Disk

Description

This command creates a read-only snapshot virtual disk for the snapshot images of a base virtual disk. To change a read-only snapshot virtual disk to a read/write virtual disk, use the `set snapVirtualDisk convertToReadWrite` command.

NOTE: You cannot use this command for a snapshot image that is used in online virtual disk copy.

Syntax

```
create snapVirtualDisk userLabel="snapVirtualDiskName"
snapImageID="snapImageID" readOnly
```

Parameters

Parameter	Description
userLabel	The name that you want to give to a snapshot virtual disk. Enclose the snapshot virtual disk name in double quotation marks (" ").
snapImageID	The alphanumeric identifier of a snapshot image that you want to add to the new snapshot virtual disk. The identifier of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • NEWEST - Use this option when you want to show the latest snapshot image created in the snapshot group. • OLDEST - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ").

Parameter	Description
`readOnly`	This parameter sets the snapshot virtual disk to read-only. This parameter is actually a Boolean; however, in the context of this command, the Boolean value is always `TRUE`. NOTE: With the `readOnly` parameter, snapshot virtual disks are not created.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

The identifier of a snapshot image has two parts separated by a colon (:):

- The name of the snapshot group.
- The identifier of the snapshot image.

For example, if you want to create a read-only virtual disk named **engData1** using the most recent snapshot image in a snapshot group that has the name **snapGroup1**, you would use this command:

```
create snapVirtualDisk userLabel="engData1" snapImageID="snapGroup1:newest"
readOnly;
```

Create Remote Replication

NOTE: This command is valid only with Fibre Channel storage arrays.

Description

This command creates both the primary virtual disk and the secondary virtual disk for a remote replication pair. This command also sets the write mode (Synchronous or Asynchronous) and the synchronization priority.

Syntax

```
create remoteReplication primary=
"primaryVirtualDiskName"
secondary="secondaryVirtualDiskName"
(remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwn="wwID")
[remotePassword="password"
syncPriority=(highest | high | medium | low |
lowest)
autoResync=(enabled | disabled)
writeOrder=(preserved | notPreserved)
writeMode=(synchronous | asynchronous)]
```

Parameters

Parameter	Description
`primary`	The name of an existing virtual disk on the local storage array that you want to use for the primary virtual disk. Enclose the primary virtual disk name in double quotation marks (" ").
`secondary`	The name of an existing virtual disk on the remote storage array that you want to use for the secondary virtual disk. Enclose the secondary virtual disk name in double quotation marks (" ").
`remoteStorageArrayName`	The name of the remote storage array. Enclose the remote storage array name in double quotation marks (" ").
`remoteStorageArrayWwn`	The World Wide Identifier (WWID) of the remote storage array. Enclose the WWID in double quotation marks (" ").
`remotePassword`	The password for the remote storage array. Use this parameter when the remote storage array is password protected. Enclose the password in double quotation marks (" ").
`syncPriority`	The priority that full synchronization has relative to host I/O activity. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`autoResync`	The settings for automatic resynchronization between the primary virtual disks and the secondary virtual disks of a remote replication pair. This parameter has these values: • `enabled`—Automatic resynchronization is turned on. You do not need to do anything further to resynchronize the primary virtual disk and the secondary virtual disk. • `disabled`—Automatic resynchronization is turned off. To resynchronize the primary virtual disk and the secondary virtual disk, you must run the `start asyncReplicationGroup` command.
`writeOrder`	The write order for data transmission between the primary virtual disk and the secondary virtual disk. Valid values are `preserved` or `notPreserved`.
`writeMode`	How the primary virtual disk writes to the secondary virtual disk. Valid values are `synchronous` or `asynchronous`.

Notes

Names can be any combination of alphanumeric characters, hyphens, and underscores. Names can have a maximum of 30 characters.

When the primary virtual disk and the secondary virtual disk are chosen, the secondary virtual disk must be of equal or of a greater size than the primary virtual disk. The RAID level of the secondary virtual disk does not have to be the same as that of the primary virtual disk.

Passwords are stored on each storage array in a management domain. If a password was not previously set, a password is not needed. The password can be any combination of alphanumeric characters with a maximum of 30 characters. (A storage array password can be defined by using the `set storageArray` command.)

Synchronization priority defines the amount of system resources that are used to synchronize the data between the primary virtual disk and the secondary virtual disk of a replication relationship. If the highest priority level is selected, the data synchronization uses the most system resources to perform the full synchronization, which decreases performance for host data transfers.

The `writeOrder` parameter applies only to asynchronous replications and makes them become part of a consistency group. Setting the `writeOrder` parameter to `preserved` causes the remote replication pair to transmit data from the primary virtual disk to the secondary virtual disk in the same order as the host writes to the primary virtual disk. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This action can require additional system overhead to maintain the buffered data, which slows operations. Setting the `writeOrder` parameter to `notPreserved` frees the system from having to maintain data in a buffer, but it requires forcing a full synchronization to make sure that the secondary virtual disk has the same data as the primary virtual disk.

Create Remote Replication Group

Description

This command creates a new, empty remote replication group on both the local storage array and the remote storage array. A remote replication group is a container that can house several replicated pairs so that they can be managed as one entity. You create a remote replication group to define the synchronization settings for all replicated pairs within the group. Each replicated pair in a remote replication group share the same synchronization settings, primary and secondary role, and write mode.

The remote replication group is associated with the local storage array and remote storage array that is used for replicating. The local storage array is the primary side of the remote replication group, while the remote storage array is the secondary side of the remote replication group. All virtual disks added to the remote replication group on the local storage array hold the primary role in the replicate relationship. Subsequently, all virtual disks added to the remote replication group on the remote storage array hold the secondary role in the replicate relationship.

Make sure that you execute the `create remote replication group` command on the local storage array. Remote replication group creation is initiated from the storage array that contains the virtual disks that hold the primary role in the replicate relationship. You use the `create remote replication`

`group` command to specify the identity of the remote storage array that contains the virtual disks that hold the secondary role in the replicate relationship.

Syntax

```
create asyncRemoteReplicationGroup userLabel="asyncRemoteReplicationGroupName"
(remoteStorageArrayName="storageArrayName" | remoteStorageArrayWwn="wwID")
interfaceType=(FC | iSCSI)
[remotePassword="password"
syncInterval=integer (minutes | hours | days)
warningSyncThreshold=integer (minutes | hours | days)
warningRecoveryThreshold=integer (minutes | hours | days)
warningThresholdPercent=percentValue
autoResync=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`userLabel`	The name of the new remote replication group that you want to create. Enclose the new remote replication group name in double quotation marks (" "). The name must be unique on the local and remote storage arrays.
`remoteStorageArrayName`	The name for the remote storage array on which you are replicating the remote replication group. Enclose the storage array name in double quotation marks (" "). The remote storage array must have the same connection type as the local storage array.
`remoteStorageArrayNameWwn`	The World Wide Identifier (WWID) of the remote storage array on which you are replicating the remote replication group. You can use the WWID instead of the storage array name to identify the storage array. Enclose the WWID in angle brackets (< >). The remote storage array must have the same connection type as the local storage array.
`interfaceType`	Specify which connection type to use: either Fibre Channel fabric or iSCSI interface. (The default is Fibre Channel.) The local storage array and the remote storage array must be connected through a proper Fibre Channel fabric or iSCSI interface.
`remotePassword`	The password for the remote storage array. Use this parameter when the remote storage array is password protected. Enclose the password in double quotation marks (" ").

Parameter	Description
`syncInterval`	Specify the length of time between automatically sending updates of modified data from the local storage array to the remote storage array. You can specify the length of time in minutes, hours, or days.
`warningSyncThreshold`	Specify the length of time to wait until a warning is triggered for cases in which the synchronization of all of the virtual disks within the remote replication group takes longer than the defined time. You can specify the length of time in minutes, hours, or days.
`warningRecoveryThreshold`	Specify the length of time to wait until a warning is triggered when the automatic data update for the point-in-time image on the remote storage array is older than the defined time. Define the threshold from the end of the previous update. You can specify the length of time in minutes, hours, or days. **NOTE:** You must set the Recovery Point Threshold to be twice as large as the synchronization interval threshold.
`warningThresholdPercent`	Specify the length of time to wait until a warning is triggered when the capacity of a replication repository virtual disk reaches the defined percentage. Define the threshold by percentage (%) of the capacity remaining.
`autoResync`	The settings for automatic resynchronization between the primary virtual disks and the secondary virtual disks of a remote replicated pair within a remote replication group. This parameter has these values: • `enabled`—Automatic resynchronization is turned on. You do not need to do anything further to resynchronize the primary virtual disk and the secondary virtual disk. • `disabled`—Automatic resynchronization is turned off. To resynchronize the primary virtual disk and the secondary virtual disk, you must run the `resume remoteReplicationGroup` command.

Notes

- The Remote Replication premium feature must be enabled and activated on the local and remote storage arrays that will be used for replication activities.
- You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

- The local and remote storage arrays must be connected through a Fibre Channel fabric or iSCSI interface.
- Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of a alphanumeric characters with a maximum of 30 characters. (You can define a storage array password by using the `set storageArray` command.)
- Depending on your configuration, there is a maximum number of remote replication groups you can create on a storage array.
- Remote replication groups are created empty and replicated pairs are added to them later on. Only replicated pairs can be added to a remote replication group. Each replicated pair is associated with exactly one remote replication group.
- The Remote Replication process is initiated at a defined synchronization interval. Periodic point-in-time images are replicated as only the changed data is copied and not the entire virtual disk.

Create Snapshot Group

Description

This command creates a new snapshot group and the associated repository virtual disk. A snapshot group contains a sequence of snapshot images of an associated base virtual disk. A snapshot group has a repository virtual disk that is used to save data for all of the snapshot images that are part of the snapshot group.

Syntax

```
create snapGroup userLabel="snapGroupName" sourceVirtualDisk="virtualDiskName"
[(repositoryVirtualDisk="repos_xxxx" | repositoryVirtualDisk=(diskGroupName
[capacity=capacityValue]) | repositoryVirtualDisk=(diskPoolName
[capacity=capacityValue]))] [repositoryFullPolicy=(failBaseWrites |
purgeSnapImages)] [rollbackPriority=(highest | high | medium | low | lowest)]
[repositoryFullLimit=percentValue] [autoDeleteLimit=numberOfSnapImages]|
[enableSchedule=(TRUE | FALSE)] [schedule (immediate | snapshotSchedule)]
```

Parameters

Parameter	Description
`userLabel`	The alphanumeric identifier (including - and _) that you want to give the new snapshot group. Enclose the snapshot group identifier in double quotation marks (" ").
`sourceVirtualDisk`	The alphanumeric identifier (including - and _) of the virtual disk that you want to use as the source for your snapshot images. Enclose the source virtual disk name in double quotation marks (" ").
`repositoryVirtualDisk`	The name of the repository virtual disk that contains the snapshot group. Enclose the snapshot repository virtual disk name in double quotation marks (" "). The repository virtual disk name is comprised of two parts:

Parameter	Description
	• The term `repos` • A four-digit numerical identifier that you assign to the repository virtual disk name
`repositoryFullPolicy`	Defines how snapshot image processing continues if the snapshot image repository virtual disk is full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or delete (purge) the snapshot images (`purgeSnapImages`). The default action is `purgeSnapImages`.
`rollBackPriority`	Determines whether system resources should be allocated to the rollback operation at the expense of system performance. A value of `high` indicates that the rollback operation is prioritized over all other host I/O. A value of `low` indicates that the rollback operation should be performed with minimal impact to host I/O. The default value is `medium`.
`repositoryFullLimit`	The percentage of repository capacity at which you receive a warning that the snapshot image repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.
`autoDeleteLimit`	The maximum number of snapshot images that you want to automatically delete if you have selected to purge the snapshot images for the repository full policy. The default value is `32`.

Notes

Each snapshot group user label must be unique. You can use any combination of alphanumeric characters, underscore (_), hyphen(-), and pound (#) for the user label. User labels can have a maximum of 30 characters.

To create a snapshot group, you must have an associated repository virtual disk in which you store the snapshot images. You can either use an existing repository virtual disk or create a new repository virtual disk. You can create the repository virtual disk when you create the snapshot group. A snapshot group repository virtual disk is an expandable virtual disk that is structured as a concatenated collection of up to 16 standard virtual disk entities. Initially, an expandable repository virtual disk has only a single element. The capacity of the expandable repository virtual disk is exactly that of the single element. You can increase the capacity of an expandable repository virtual disk by attaching additional standard virtual disks to it. The composite expandable repository virtual disk capacity then becomes the sum of the capacities of all of the concatenated standard virtual disks.

A snapshot group has a strict ordering of snapshot images based on the time that each snapshot image is created. A snapshot image that is created after another snapshot image is a successor relative to that

other snapshot image. A snapshot image that is created before another snapshot image is a predecessor relative to that other one.

A snapshot group repository virtual disk must satisfy a minimum capacity requirement that is the sum of the following:

- 32 MB to support fixed overhead for the snapshot group and for copy-on-write processing.
- Capacity for rollback processing, which is 1/5000th of the capacity of the base virtual disk.

The minimum capacity is enforcement by the RAID controller module firmware and the MD storage management software.

When a you first create a snapshot group, it does not contains any snapshot images. When you create snapshot images, you add the snapshot images to a snapshot group. Use the `create snapImage` command to create snapshot images and add the snapshot images to a snapshot group.

A snapshot group can have one of these states:

Optimal	The snapshot group is operating normally.
Full	The snapshot group repository is full. Additional copy-on-write operations can not be performed. This state is possible only for snapshot groups that have the Repository Full policy set to Fail Base Writes. Any snapshot group in a Full state causes a Needs-Attention condition to be posted for the storage array.
Over Threshold	The snapshot group repository virtual disk usage is at or beyond its alert threshold. Any snapshot group in this state causes a Needs-Attention condition to be posted for the storage array.
Failed	The snapshot group has encountered a problem that has made all snapshot images in the snapshot group unusable. For example, certain types of repository virtual disk failures can cause a Failed state. To recover from a Failed state use the `revive snapGroup` command.

Create Snapshot Image

Description

This command creates a new snapshot image in one or more existing snapshot groups. Before you can create a snapshot image, you must first have at least one snapshot group into which you can place the snapshot image. To create a snapshot group use the `create snapGroup` command.

Syntax

```
create snapImage (snapGroup="snapGroupName" |
snapGroups=("snapGroupName1"... "snapGroupName2"))
```

Parameters

Parameter	Description
`snapGroup` or `snapGroups`	The name of the snapshot group that will contain the snapshot image. Enclose the name of the snapshot group in double quotation marks (" "). If you enter more than one snapshot group name, enclose each of the snapshot group names in double quotation marks (" "). Enclose all of the snapshot group names in parentheses.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

A snapshot image is a logical point-in-time image of the contents of an associated base virtual disk. The snapshot image is created instantly and records the state of the base virtual disk at that moment. Every snapshot image is created in the context of exactly one snapshot group. A snapshot group is a sequence of snapshot images of the associated base virtual disk. A snapshot group has one repository virtual disk used to save all of the data from the snapshot images. The snapshot images in a snapshot group have a specific order. The specific order of the snapshot images enables you to manage the snapshot images, such as restoring a specific snapshot image to the base virtual disk or deleting the snapshot images that you no longer need.

The result of creating a snapshot image of a consistency group is a snapshot image of every member virtual disk of the consistency group.

Create Snapshot (Legacy) Virtual Disk

Description

This command creates a snapshot virtual disk. When you use this command, you can define the snapshot virtual disk in one of three ways:

- User-defined physical disks
- User-defined disk group
- User-defined number of physical disks for the snapshot virtual disk

If you choose to define a number of physical disks, the RAID controller module firmware chooses which physical disks to use for the snapshot (legacy) virtual disk.

Syntax (User-Defined Physical Disks)

```
create snapshotVirtualDisk baseVirtualDisk="baseVirtualDiskName"
[repositoryRAIDLevel=(0 | 1 | 5 | 6)]
[repositoryPhysicalDisks=(enclosureID1,slotID1 ... enclosureIDn,slotIDn)]
[repositoryDiskGroupUserLabel="repositoryDiskGroupName"]
[enclosureLossProtect=(TRUE | FALSE)]
[freeCapacityArea=freeCapacityIndexNumber]
[userLabel="snapshotVirtualDiskName"]
[warningThresholdPercent=percentValue]
```

```
[repositoryPercentOfBase=percentValue]
[repositoryUserLabel="repositoryName"]
[repositoryFullPolicy=(failBaseWrites | failSnapshot)]
[enableSchedule=(TRUE | FALSE)]
[schedule=(immediate | snapshotSchedule)]
```

Syntax (User-Defined Number Of Physical Disks)

```
create snapshotVirtualDisk baseVirtualDisk="baseVirtualDiskName"
[repositoryRAIDLevel=(0 | 1 | 5 | 6)]
[repositoryPhysicalDiskCount=numberOfPhysicalDisks ]
[repositoryDiskGroupUserLabel="repositoryDiskGroupName"]
[physicalDiskMediaType=(HDD | SSD | unknown | allMedia)]
[physicalDiskType=(SAS)]
[enclosureLossProtect=(TRUE | FALSE)]
[userLabel="snapshotVirtualDiskName"]
[warningThresholdPercent=percentValue]
[repositoryPercentOfBase=percentValue]
[repositoryUserLabel="repositoryName"]
[repositoryFullPolicy=(failBaseWrites | failSnapshot)]
[enableSchedule=(TRUE | FALSE)]
[schedule=(immediate | snapshotSchedule)]
```

Syntax (Existing Repository Disk Group)

```
create snapshotVirtualDisk baseVirtualDisk="baseVirtualDiskName"
[repositoryDiskGroup="repositoryDiskGroupName"]
[repositoryUserLabel="repositoryName"]
[freeCapacityArea=freeCapacityIndexNumber]
[userLabel="snapshotVirtualDiskName"]
[warningThresholdPercent=percentValue]
[repositoryPercentOfBase=percentValue]
[repositoryFullPolicy=(failBaseWrites | failSnapshot)]
[enableSchedule=(TRUE | FALSE)]
[schedule=(immediate | snapshotSchedule)]
```

Parameters

Parameter	Description
`baseVirtualDisk`	The name of the base virtual disk from which you want to take a snapshot (legacy). Enclose the base virtual disk name in double quotation marks (" ").
`repositoryRAIDLevel`	Use this parameter when you create a new disk group. The RAID level for the snapshot (legacy) repository disk group. Valid values are `0`,`1`, `5`, or `6`.
`repositoryPhysicalDisks`	Use this parameter when you create a new disk group. The physical disks that you want to assign to the snapshot (legacy) repository disk group. Specify the enclosure ID value and the slot ID value for each physical disk that you assign to the snapshot (legacy) repository virtual disk. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. Enclose the

Parameter	Description
	enclosure ID values and the slot ID values in parentheses.
repositoryPhysicalDiskCount	Use this parameter when you create a new disk group. The number of unassigned physical disks that you want to use for the snapshot (legacy) repository disk group.
repositoryDiskGroupUserLabel	Use this parameter when you create a new disk group. The name of a new disk group to be used for the repository virtual disk. Enclose the repository disk group name in double quotation marks (" ").
repositoryDiskGroup	The name of an existing disk group where you want to place the repository virtual disk. Use this parameter if you do not want to put the repository virtual disk in the same disk group as the base virtual disk. The default is to use the same disk group for both the base virtual disk and the repository virtual disk. Enclose the name of the repository disk group in double quotation marks (" ").
userLabel	The name that you want to give to the snapshots (legacy) virtual disk. If you do not want to provide a name, the CLI creates a name using the base virtual disk user label that you provide.
enclosureLossProtect	The setting to enforce enclosure loss protection when you create the snapshot (legacy) repository virtual disk. To enforce enclosure loss protection, set this parameter to TRUE. The default setting is FALSE.
physicalMediaType	The type of physical disk media that you want to use for the disk group. Valid physical disk media consists of: • HDD—Use this option when you have hard physical disks in the expansion enclosure. • SSD—Use this option when you have solid state physical disks in the expansion enclosure. • unknown—Use if you are not sure what types of physical disk media are in the expansion enclosure. • allMedia—Use this option when you want to use all types of physical disk media that are in the expansion enclosure.

Parameter	Description
	Use this parameter when you use the `repositoryPhysicalDiskCount` parameter. You must use this parameter when you have more than one type of physical disk media in your storage array.
`physicalDiskType`	The type of physical disk that you want to use in the virtual disk. You cannot mix physical disk types. You must use this parameter when you have more than one type of physical disk in your storage array. The valid physical disk type is `SAS`. You must specify a physical disk type. Use this parameter when you use the `repositoryPhysicalDiskCount` parameter.
`freeCapacityArea`	The index number of the free space in an existing disk group that you want to use to create the snapshot (legacy) repository virtual disk. Free capacity is defined as the free capacity between existing virtual disks in a disk group. For example, a disk group might have the following areas: virtual disk 1, free capacity, virtual disk 2, free capacity, virtual disk 3, free capacity. To use the free capacity following virtual disk 2, you specify: `freeCapacityArea=2` Use the `show diskGroup` command to determine if free capacity area exists.
`warningThresholdPercent`	The percentage of repository capacity at which you receive a warning that the snapshot (legacy) repository is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `50`.
`repositoryPercentOfSource`	The size of the snapshot (legacy) repository virtual disk as a percentage of the source virtual disk. Use integer values. For example, a value of `40` means 40 percent. The default value is `20`.
`repositoryUserLabel`	The name to give to the snapshot (legacy) repository virtual disk. Enclose the snapshot (legacy) repository virtual disk name in double quotation marks (" ").
`repositoryFullPolicy`	Specifies how snapshot (legacy) processing continues if the snapshot (legacy) repository is full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or fail the snapshot

Parameter	Description
	(legacy) virtual disk (`failSnapShot`). The default value is `failSnapShot`.
`enableSchedule`	Turns on or off the ability to schedule a snapshot (legacy) operation. To turn on snapshot (legacy) scheduling, set this parameter to `TRUE`. To turn off snapshot (legacy) scheduling, set the parameter to `FALSE`.
`schedule`	Use this parameter to schedule a snapshot (legacy) operation. You can use one of these options for setting a schedule for a snapshot (legacy) operation: • `immediate` • `startDate` • `scheduleDay` • `startTime` • `scheduleInterval` • `endDate` • `noEndDate` • `timesPerDay` • `timeZone` NOTE: For more information, see Notes below.

Notes

The virtual disk that you are taking a snapshot (legacy) of must be a standard virtual disk in the storage array. The maximum number of snapshot (legacy) virtual disks that you can create is one-half of the total number of virtual disks that are supported by a RAID controller module.

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

One technique for naming the snapshot (legacy) virtual disk and the snapshot (legacy) repository virtual disk is to add a hyphenated suffix to the original base virtual disk name. The suffix distinguishes between the snapshot (legacy) virtual disk and the snapshot (legacy) repository virtual disk. For example, if you have a base virtual disk with a name of **Engineering Data**, the snapshot (legacy) virtual disk can have a name of **Engineering Data-S1**, and the snapshot (legacy) repository virtual disk can have a name of **EngineeringData- R1**.

If you do not choose a name for either the snapshot (legacy) virtual disk or the snapshot (legacy) repository virtual disk, the MD storage management software creates a default name by using the base virtual disk name. An example of the snapshot (legacy) virtual disk name that the RAID controller modules might create is, if the base virtual disk name is **aaa** and does not have a snapshot (legacy) virtual disk, the default snapshot (legacy) virtual disk name is **aaa – 1**. If the base virtual disk already has $n - 1$ number of snapshot (legacy) virtual disks, the default name is **aaa –** n. An example of the snapshot (legacy)

repository virtual disk name that the RAID controller module might create is, if the base virtual disk name is **aaa** and does not have a snapshot (legacy) repository virtual disk, the default snapshot (legacy) repository virtual disk name is **aaa – R1**. If the base virtual disk already has *n* – 1 number of snapshot (legacy) repository virtual disks, the default name is **aaa – R***n*.

If you do not specify the unconfigured space or free space, the snapshot (legacy) repository virtual disk is placed in the same disk group as the base virtual disk. If the disk group where the base virtual disk resides does not have enough space, this command fails.

Enclosure Loss Protection

When you assign the physical disks, if you set the `enclosureLossProtect` parameter to `TRUE` and have selected more than one physical disk from any one enclosure, the storage array returns an error. If you set the `enclosureLossProtect` parameter to `FALSE`, the storage array performs operations, but the disk group that you create might not have enclosure loss protection.

When the RAID controller module firmware assigns the physical disks, if you set the `enclosureLossProtect` parameter to `TRUE`, the storage array returns an error if the RAID controller module firmware cannot provide physical disks that result in the new disk group having enclosure loss protection. If you set the `enclosureLossProtect` parameter to `FALSE`, the storage array performs the operation even if it means the disk group might not have enclosure loss protection.

Scheduling Snapshots (Legacy)

You can use the `enableSchedule` and the `schedule` parameters to schedule automatic snapshots (legacy). Automatic snapshots (legacy) can be scheduled daily, weekly, or monthly (by day or by date). The `enableSchedule` parameter turns on or off the ability to schedule snapshots (legacy). When you enable scheduling, you use the `schedule` parameter to define when you want the snapshots (legacy) to occur.

This list explains how to use the options for the `schedule` parameter:

immediate — As soon as you enter the command, a snapshot (legacy) virtual disk is created and a copyon- write operation begins.

startDate — A specific date on which you want to create a snapshot (legacy) virtual disk and perform a copy-on-write operation. The format for entering the date is `MM:DD:YY`. If you do not provide a start date, the current date is used. An example of this option is `startDate=06:27:11`.

scheduleDay — A day of the week on which you want to create a snapshot (legacy) virtual disk and perform a copy-on-write operation. The values that you can enter are: `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, `sunday`, and all. For example: `scheduleDay=wednesday`.

startTime — The time of a day you want to create a snapshot (legacy) virtual disk and start performing a copy-on-write operation. The format for entering the time is `HH:MM`, where `HH` is the hour and `MM` is the minute past the hour. Use a 24-hour clock. For example: `startTime=14:27`, which corresponds to 2:27 PM.

scheduleInterval — Amount of time, in minutes, that defines a minimum between two copy-on-write operations. It is possible for you to create a schedule in which you have overlapping copy-on-write operations because of the duration of a copy

operation. You can ensure that copy-on-write operations maintain a correct interval by using this option. The maximum value for the `scheduleInterval` option is 1440 minutes. For example: `scheduleInterval=180`.

endDate — Specific date to stop creating a snapshot (legacy) virtual disk and end the copy-on-write operations. The format for entering the date is `MM:DD:YY`, where `MM` is the month, `DD` the day and `YY` the year. For example: `endDate=11:26:11`.

timesPerDay — The number of times you want the snapshot schedule to run in a day. For example, `timesPerDay=4`.

timeZone — Use this parameter to define the time zone in which the storage array is operating. You can define the time zone in one of two ways:

- `GMT±HH:MM`—The time zone offset from Greenwich Mean Time (GMT), in hours and minutes. For example, GMT-06:00 is the central time zone in the United States.
- `Text string`—A standard time zone text string, in quotation marks. For example, "USA/Chicago" or "Germany/Berlin". Time zone text strings are case sensitive. If you enter an incorrect text string GMT is used.

The code string for defining a schedule is similar to these examples:

```
enableSchedule=true schedule startTime=14:27

enableSchedule=true schedule scheduleInterval=180

enableSchedule=true schedule timeZone=GMT-06:00

enableSchedule=true schedule timeZone="USA/Chicago"
```

If you also use the `scheduleInterval` option, the lowest value of the `timesPerDay` option and the `scheduleInterval` option is used. The firmware calculates an integer value for the `scheduleInterval` option by dividing 1440 by the `scheduleInterval` option value that you set. For example, 1440/180 = 8. The firmware then compares the `timesPerDay` integer value with the calculated `scheduleInterval` integer value and uses the smaller value.

To remove a schedule, use the `delete virtualDisk` command with the schedule parameter. The `delete virtualDisk` command with the schedule parameter deletes only the schedule, not the snapshot (legacy) virtual disk.

Create Snapshot Virtual Disk

Description

This command creates a snapshot virtual disk for snapshot images of a base virtual disk. You can assign the new snapshot virtual disk to an existing repository disk group or disk pool, or you can create a new repository disk group or disk pool if one does not already exist.

NOTE: You cannot use this command for a snapshot image that is used in online virtual disk copy.

Syntax

```
create snapVirtualDisk userLabel="snapVirtualDiskName" snapImageID="snapImageID"
[repositoryVirtualDisk="repos_xxxx" |
repositoryFullLimit=percentValue]
```

Parameters

Parameter	Description
`userLabel`	The alphanumeric identifier (including - and _) that you want to give to a snapshot virtual disk. Enclose the snapshot virtual disk name in double quotation marks (" ").
`snapImageID`	The alphanumeric identifier of a snapshot image that you want to add to the new snapshot virtual disk. The identifier of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ").
`repositoryVirtualDisk`	The alphanumeric identifier of the repository virtual disk has the snapshot image. All repository identifiers have this form: `repos_xxxx` where xxxx represents a four digit numerical value. Enclose the snapshot virtual disk name in double quotation marks (" ").
`repositoryFullLimit`	The percentage of repository capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

The identifier of a snapshot image has two parts separated by a colon (:):

- The name of the snapshot group.
- The identifier of the snapshot image.

For example, if you want to create a snapshot virtual disk named **snapData1**, using the most recent snapshot image in a snapshot group that has the name **snapGroup1**, with a maximum fill limit of 80 percent for the repository virtual disk, you would use this command:

```
create snapVirtualDisk userLabel="snapData1" snapImageID="snapGroup1:newest"
repositoryVirtualDisk="repos_1234" repositoryFullLimit=80;
```

The repository virtual disk identifier is automatically created by the MD storage management software and the firmware when you create a new snapshot group. You cannot rename the repository virtual disk because renaming the repository virtual disk will break the linkage with the snapshot images.

Create SSD Cache

Description

This command creates a read cache for a storage array using Solid State Disks (SSDs). Using high performance SSDs to cache read data improves the application I/O performance and response times, and delivers sustained performance improvement across different workloads, especially for high-IOP workloads. SSD cache maximizes the use of expensive fast SSDs.

SSD cache works in addition to the primary cache in the RAID controller module DRAM. With RAID controller module cache, the data is stored in DRAM after a host read. With SSD cache, the data is copied from user specified base virtual disks, and then cached on SSDs.

Syntax

```
create ssdCache userLabel="ssdCacheName"
physicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
[updateExistingVirtualDisks=(TRUE|FALSE)]
```

Parameters

Parameter	Description
userLabel	The alphanumeric identifier (including - and _) that you want to name the new SSD cache. Enclose the identifier in double quotation marks (" "). You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound

Parameter	Description
	(#) for the identifier. Identifiers can have a maximum of 30 characters.
`physicalDisks`	The physical disks that you want to use to create the SSD cache. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each physical disk that you want to assign to the SSD cache. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each physical disk that you assign to the SSD cache. Enclosure ID values are `0` to `99`. Drawer ID values are `0` to `4`. Slot ID values are `0` to `31`. Enclose the enclosure ID values, the drawer ID values, and the slot ID values in parentheses.
`updateExistingVirtualDisks`	This optional parameter specifies whether the SSD cache should be enabled for all existing virtual disks in the storage array. To enable the SSD cache for all existing virtual disks, set this parameter to `TRUE`. If you want to enable the SSD cache for individual virtual disks later, set this parameter to `FALSE`. The default value is `TRUE`.

Notes

A storage array can have only one SSD cache.

Only virtual disks created using hard disk physical disks can use the SSD cache. You cannot enable SSD cache on snapshot (legacy) virtual disks or snapshot images.

SSDs do not currently support self encrypting disk (SED). Therefore, virtual disks that have SED enabled cannot use the SSD cache.

If all of the SSDs in the SSD cache are data assurance (DA)-capable and the DA premium feature is enabled, DA is automatically enabled for the SSD cache and cannot be disabled. In addition, you cannot add non-DA capable SSDs to a DA-enabled SSD cache.

Create Storage Array Security Key

Description

This command creates a new security key for a storage array that has Self-Encrypting Disk (SED) physical disks. This command also sets the security definitions and sets the state to **Security Enabled**.

NOTE: Before you create a storage array security key, you must set the password for the storage array. Use the `setStorageArray` command to set the password for the storage array.

Syntax

```
create storageArray securityKey  [keyIdentifier="keyIdentifierString"]
[passPhrase="passPhraseString"]  [file="fileName"] [commitSecurityKey=(TRUE |
FALSE)]
```

Parameters

Parameter	Description
`keyIdentifier`	A character string that you can read that is a wrapper around a security key.
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file.
`file`	The file path and the file name to which you want to save the security key. For example, **file="C:\Program Files\CLI\sup\seckey.slk"** **NOTE:** You must add a file extension of **.slk** to the end of the file name.
`commitSecurityKey`	This parameter commits the security key identifier to the storage array for all SES drives as well as the controllers. After the security key identifier is committed, a key is required to read data or write data. The data can only be read or changed by using a key, and the drive can never be used in a nonsecure mode without rendering the data useless or totally erasing the drive.

Additional Information

To use this command successfully, you need to have enough SES drives to create atleast one disk group. The RAID controller module firmware creates a lock that restricts access to the SES drives. SES drives have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all SES drives that exist within the storage array.

You can have a storage array configuration with more than one set of encrypted disk groups. Each disk group can have a unique security key. The character string generated by the `keyIdentifier` parameter is a string that you can read and that enables you to identify the security key that you need. You can create a keyIdentifer by using one of these methods:

- You can enter up to 189 alphanumeric characters for a key identifier. The key identifier cannot have these characters:
  - White spaces
  - Punctuation
  - Symbols
- You can leave out this parameter.

Additional characters are automatically generated and appended to the end of the string that you enter for the key identifier. If you do not enter any string for the `keyIdentifier` parameter, the key identifier consists of only the characters that are automatically generated.

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

If your pass phrase does not meet these criteria, you receive an error message and you must retry the command.

Create Virtual Disk Copy

Description

This command creates a virtual disk copy and starts the virtual disk copy operation.

CAUTION: Before you create a new copy of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk and (if applicable, the target disk) to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: See Preparing Host Servers To Create A Virtual Disk Copy.

NOTE: Removing the drive letter of the associated virtual disk(s) in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the virtual copy.

NOTE: You can have a maximum of eight virtual disk copies in progress at one time. If you try to create more than eight virtual disk copies at one time, the RAID controller modules return a status of Pending until one of the virtual disk copies that is in progress finishes and returns a status of Complete.

This command creates virtual disk copies in two ways:

- Virtual disk copy without snapshot (legacy)
- Virtual disk copy with snapshot (legacy)

If you use `virtual disk copy without snapshot (legacy)`, you cannot write to the source virtual disk until the copy operation is complete. If you want to be able to write to the source virtual disk before the copy operation is complete, use `virtual disk copy with snapshot (legacy)`. You can select `virtual disk copy with snapshot (legacy)` through the optional parameters in the command syntax.

Syntax

```
create virtualDiskCopy source="sourceName"
target="targetName"
[copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)]
copyType=(offline | online)
```

Parameters

Parameter	Description
`source`	Name of an existing virtual disk to use as the source virtual disk. You must put quotation marks (" ") around the source virtual disk name. NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.
`target`	Name of an existing virtual disk to use as the target virtual disk. You must put quotation marks (" ") around the target virtual disk name. NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.
`copyPriority`	Specifies the priority that a virtual disk copy has relative to host I/O activity. Valid entries are `highest`, `high`, `medium`, `low`, or `lowest`. NOTE: `CopyPriority` defines the amount of system resources used to copy the data between the source virtual disk and target virtual disk of a virtual disk-copy pair. If you select the highest priority level, the virtual disk copy uses the most system resources to perform virtual disk copy, which decreases performance for host data transfers.
`targetReadOnlyEnabled`	Specifies whether the target virtual disk is write enabled or read only. To be able to write to the target virtual disk, set this parameter to `FALSE`. To prevent writing to the target virtual disk, set this parameter to `TRUE`.
`copyType`	Creates a virtual disk copy with or a snapshot (legacy). Creating a virtual disk copy with a snapshot (legacy) enables you to continue to write to the source virtual disk while creating the virtual disk copy. To create a virtual disk copy with a

Parameter	Description
	snapshot (legacy), set this parameter to online. To create a virtual disk copy without a snapshot (legacy), set this parameter to `offline`. If this parameter is not used, the virtual disk copy is created without a snapshot (legacy).

Deactivate Remote Replication (Legacy)

NOTE: This command valid only with Fibre Channel storage arrays.

Description

This command deactivates the Remote Replication (legacy) premium feature, disassembles the replication repository virtual disk, and releases the RAID controller module owner of the secondary virtual disk. The RAID controller module host port that is dedicated to the secondary virtual disk is available for host data transfers.

Syntax

```
deactivate storageArray feature=remoteReplication
```

Parameters

None.

Delete Consistency Group

Description

This command deletes a snapshot consistency group in two ways:

- You can delete both the consistency group and the repository virtual disks contained by the consistency group.
- You can delete only the consistency group and leave the repository virtual disks that are contained by the consistency group intact.

Syntax

```
delete consistencyGroup [snapGroupName]
[deleteRepositoryMembers=(TRUE | FALSE)]]
```

Parameter

Parameter	Description
`snapConsistencyGroup`	The name of the snapshot consistency group that you want to delete. Enclose the snapshot consistency group name in square brackets ([ ]). If the snapshot consistency group name has special

Parameter	Description
	characters, you must enclose the snapshot consistency group name in double quotation marks (" ").
deleteRepositoryMembers	The setting to delete on or retain the repository virtual disks. To delete the repository virtual disks, set this parameter to TRUE. To retain the repository virtual disks, set this parameter to FALSE. The default setting is FALSE.

Delete Consistency Group Snapshot Image

Description

This command deletes the snapshot images in a consistency group.

Syntax

```
delete cgSnapImage consistencyGroup="consistencyGroupName"
[deleteCount=numberOfSnapImages]
[retainCount=numberOfSnapImages]
[ignoreSnapVolume=(TRUE | FALSE)]
```

Parameter

Parameter	Description
consistencyGroup	The name of the consistency group from which you want to delete the snapshot images. Enclose the consistency group name in double quotation marks (" ").
deleteCount	The number of snapshot images that you want to delete from the consistency group. Use integer values. This parameter deletes the oldest snapshot image first and continues to delete the oldest snapshot images until reaching the number that you enter.
retainCount	The number of snapshot images that you want to keep in the consistency group. Use integer values. This parameter keeps the most recent snapshot images in the consistency group.
ignoreSnapVirtualDisk	Determines whether the associated consistency group snapshot virtual disk is kept or deleted. This parameter applies only if the consistency group snapshot image is associated with a consistency group snapshot virtual disk. To keep the snapshot virtual disk, set this parameter to TRUE. To delete

Parameter	Description
	the snapshot virtual disk, set this parameter to FALSE. The default value is FALSE.

Notes

If the snapshot images cannot be deleted for all of the relevant consistency groups, the operation fails and none of the snapshot images are deleted.

When you delete a consistency group snapshot image that is associated with a consistency group snapshot virtual disk, the corresponding snapshot virtual disk member in the consistency group snapshot virtual disk transition to the Stopped state. A snapshot virtual disk member in the Stopped state no longer has a relationship to the snapshot group of the deleted snapshot image. However, a snapshot virtual disk member in the Stopped state keeps its relationship to its consistency group snapshot virtual disk.

Delete Consistency Group Snapshot Virtual Disk

Description

This command deletes the snapshot virtual disk of a consistency group. You can also delete the repository members.

Syntax

```
delete cgSnapVirtualDisk ["snapVirtualDiskName"]
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
cgSnapVirtualDisk	The name of the consistency group snapshot virtual disk that you want to delete. Enclose the name of the consistency group snapshot virtual disk in double quotation marks (" ") inside square brackets ([ ]).
deleteRepositoryMembers	The parameter to save or delete the member virtual disks. To save the member virtual disks, set this parameter to TRUE. To delete the member virtual disks, set this parameter to FALSE. The default value is TRUE.

Delete Disk Group

Description

This command deletes an entire disk group and its associated virtual disks.

CAUTION: All data in the disk group is lost as soon as you run this command.

Syntax

```
delete diskGroup [diskGroupNumber]
```

Parameter

Parameter	Description
`diskGroup`	Number of the disk group to delete. You must put brackets ([ ]) around the disk group number.

Delete Disk Pool

Description

This command deletes a disk pool.

Syntax

```
delete diskPool [diskPoolName]
```

Parameter

Parameter	Description
`diskPoolName`	The alphanumeric identifier of the disk pool (including – and _) that you want to delete. Enclose the disk pool name in square brackets ([ ]). If the disk pool name has special characters, you also must enclose the host group name in double quotation marks (" ").

Notes

If the disk pool contains virtual disks, the virtual disks are deleted along with the disk pool.

Delete Host

Description

This command deletes a host.

Syntax

```
delete host [hostName]
```

Parameter

Parameter	Description
`host`	Name of the host to delete. You must put brackets ([ ]) around the host name. If the host name has special characters, you must also put quotation marks (" ") around the host name. NOTE: A host is a system that is attached to the storage array and accesses the virtual disks on the storage array through its HBA host ports.

Delete Host Group

Description

This command deletes a host group.

CAUTION: This command deletes all of the host definitions in the host group.

Syntax

```
delete hostGroup [hostGroupName]
```

Parameter

Parameter	Description
`hostGroup`	Name of the host group to delete. You must put brackets ([ ]) around the host group name. If the name of the host group has special characters, you must also put quotation marks (" ") around the host group name. NOTE: A host group is an optional topological element that is a collection of hosts that share access to the same virtual disks. The host group is a logical entity.

Delete Host Port

Description

This command deletes an HBA host port identification. The identification is a software value that represents the physical HBA host port to the RAID controller module. By deleting the identification, the RAID controller module no longer recognizes instructions and data from the HBA host port.

Syntax

```
delete hostPort [hostPortName]
```

Parameter

Parameter	Description
`hostPort`	Name of the HBA host port to delete. You must put brackets ([ ]) around the name of the HBA host port. NOTE: An HBA host port is a physical connection on a host bus adapter that resides within a host system. An HBA host port provides a host access to the virtual disks in a storage array. If the host bus adapter has only one physical connection (one host port), the terms HBA host port and host bus adapter are synonymous.

Example

```
-c "delete host [\"job2900\"];"
```

Delete iSCSI Initiator

Description

This command deletes a specific iSCSI initiator object.

Syntax

```
delete iscsiInitiator (["iSCSI-ID "| "name"])
```

Parameters

Parameters	Description
iSCSI-ID	The identifier of the iSCSI initiator that you want to delete. Enclose the name in double quotation marks (" ").
name	The name of the iSCSI initiator that you want to delete. Enclose the name in double quotation marks (" ").

Example

```
-c "delete iscsiInitiator [\"job29002\"];"
```

Delete Snapshot Group

CAUTION: Possible damage to the storage array configuration – All of the data in the snapshot group is lost as soon as you run this command.

Description

This command deletes an entire snapshot group and optionally the associated member virtual disks.

Syntax

```
delete snapGroup ["snapGroupName"] [deleteRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group that you want to delete. Enclose the snapshot group identifier in double quotation marks (" ") inside square brackets ([ ]).
`deleteRepositoryMembers`	The parameter to save or delete the member virtual disks. To save the member virtual disks, set this parameter to `FALSE`. To delete the member virtual disks, set this parameter to `TRUE`. The default value is `FALSE`.

Notes

You can delete a snapshot group if it is empty or if it contains snapshot images. All of the snapshot images in the snapshot group are deleted along with the snapshot group. If any existing snapshot image within the snapshot group has an associated view, each view is stopped and detached from the snapshot image. When you delete a snapshot group the associated repository virtual disk is also deleted. By default, all member virtual disks in the repository virtual disk are retained as unused, unmapped standard virtual disks. To remove the member virtual disks set the `deleteRepositoryMembers` parameter to `TRUE`, or do not use this parameter. To keep the member virtual disks, set the `deleteRepositoryMembers` parameter to `FALSE`.

Delete Snapshot Image

Description

This command deletes one or more snapshot images from a snapshot group.

Syntax

```
delete snapImage (snapGroup="snapshotGroupName" |
snapGroups=("snapshotGroupName1"... "snapshotGroupNamen"))
(deleteCount=numberOfSnapshotImages |
retainCount=numberOfSnapshotImages |
```

```
ignoreSnapVirtualDisk=(TRUE | FALSE) |
snapImageID=OLDEST)
```

Parameters

Parameter	Description
`snapGroup` or `snapGroups`	The name of the snapshot group that has the snapshot image that you want to delete. Enclose the name of the snapshot group in double quotation marks (" "). If you enter more than one snapshot group name, enclose each of the snapshot group names in double quotation marks (" "). Enclose all of the snapshot group names in parentheses. If you do not use any other parameters with the `snapGroup` parameter or `snapGroups` parameter, then by default the oldest snapshot image is deleted.
`deleteCount`	The number of snapshot images that you want to delete from the snapshot group. Use integer values. This parameter deletes the oldest snapshot image first and continues to delete the oldest snapshot images until reaching the number that you enter. If the number that you enter is greater than the actual number of all of the snapshot images in the snapshot group, all of the snapshot images will be deleted. The snapshot group is left empty.
`retainCount`	The number of snapshot images that you want to keep in the snapshot group. Use integer values. This parameter keeps the most recent snapshot images in the snapshot group and deletes older snapshot images. If the number of existing snapshot images in the snapshot group is less than the number that you enter, none of the snapshot images are deleted.
`ignoreSnapVirtual Disk`	Use this parameter to make sure that you do not delete a snapshot image that has a snapshot virtual disk associated with the snapshot image. You can use one of these values: • `TRUE`—Use this value to delete the snapshot image even though the snapshot image has an associated snapshot virtual disk. • `FALSE`—Use this value to keep the snapshot image when the snapshot image has an associated snapshot virtual disk. The default value is `TRUE`.

Parameter	Description
`snapImageID`	The `snapImageID` parameter accepts only the `OLDEST` option. This parameter deletes the earliest snapshot image created.

Notes

You can delete the oldest snapshot image from a snapshot group repository virtual disk. The definition of a snapshot image that you delete is removed from the system. The space occupied by the snapshot image that you delete from the snapshot group repository virtual disk is released and made available for reuse within the snapshot group.

Any snapshot virtual disks that exist for a snapshot image transition to the Stopped state when the snapshot image is deleted.

This command will not run when the RAID controller module is in **Lockdown** mode.

Delete Snapshot (Legacy) Virtual Disk

Description

This command deletes one or more snapshot (legacy) virtual disks or snapshot (legacy) repository virtual disks. You can also use this command to remove schedules for creating snapshots (legacy).

CAUTION: Possible damage to the storage array configuration – All of the data in the virtual disk is lost as soon as you run this command.

Syntax

```
delete (virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ... virtualDiskNameN])
[schedule]
```

Parameters

Parameter	Description
`virtualDisk` or `virtualDisks`	The name of the snapshot (legacy) virtual disk that you want to delete. You can enter more than one snapshot (legacy) virtual disk name. Enclose the snapshot (legacy) virtual disk name in square brackets ([ ]). If the snapshot (legacy) virtual disk name has special characters, you also must enclose the snapshot (legacy) virtual disk name in double quotation marks (" ").
`schedule`	This parameter deletes the schedule for a specific snapshot (legacy) virtual disk. Only the schedule is deleted, the snapshot (legacy) virtual disk remains.

Delete Snapshot Virtual Disk

Description

This command deletes a snapshot virtual disk and optionally the associated snapshot repository members.

NOTE: You cannot use this command for snapshot images involved in online virtual disk copy.

Syntax

```
delete snapVirtualDisk ["snapVirtualDiskName"]
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`snapVirtualDisk`	The name of the snapshot virtual disk that you want to delete. Enclose the snapshot virtual disk identifier in double quotation marks (" ") inside square brackets ([ ]).
`deleteRepositoryMembers`	The parameter to save or delete the repository members. To save the repository members, set this parameter to `FALSE`. To delete the repository members, set this parameter to `TRUE`. The default value is `TRUE`. If you do not use this parameter the repository members are automatically deleted.

Delete Snapshot Virtual Disk or Delete Snapshot Virtual Disk Schedule

Description

This command deletes one or more snapshot virtual disks or snapshot repository virtual disks. You can also use this command to remove schedules for creating snapshots.

CAUTION: All of the data in the virtual disk is lost as soon as you run this command. If you are using the schedule option to delete the snapshot schedule, ensure that you do not inadvertently delete the virtual disk.

Syntax

```
delete (virtualDisk [virtualDiskName] | virtualDisks [
virtualDiskName1... virtualDiskNameN]) [schedule]
```

Parameters

Parameter	Description
`virtualDisk` or `virtualDisks`	The name of the snapshot virtual disk to delete. You can enter more than one snapshot virtual disk name. Enclose the snapshot virtual disk name in square brackets ([ ]). If the snapshot virtual disk name has special characters, you must also enclose the snapshot virtual disk name in double quotation marks (" ").
`schedule`	Deletes the schedule for a specific snapshot virtual disk. Only the schedule is deleted, the snapshot virtual disk remains.

Delete SSD Cache

Description

This command deletes the SSD cache. All data in the SSD cache is purged.

Syntax

```
delete ssdCache [ssdCacheName]
```

Parameter

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache that you want to delete. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Delete Virtual Disk

Description

This command deletes one or more standard virtual disks or snapshot virtual disks, or snapshot repository virtual disks.

CAUTION: All the data in the virtual disk is lost as soon as you run this command.

Syntax

```
delete (allVirtualDisks | virtualDisk [virtualDiskName]
|virtualDisks [virtualDiskName1... virtualDiskNameN])
removeDiskGroup=(TRUE | FALSE)
```

Parameters

Parameter	Description
`allVirtualDisks`	Deletes all virtual disks in a storage array. NOTE: Using the `allVirtualDisks` parameter deletes virtual disks until all are removed or until an error is encountered. If an error is encountered, this command does not attempt to delete the remaining virtual disks.
`virtualDisk or virtualDisks`	Name of the virtual disk to delete. You can enter more than one virtual disk name. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has special characters, you must also put quotation marks (" ") around the virtual disk name.
`removeDiskGroup`	Deleting the last virtual disk in a disk group does not delete the disk group. You can have a standalone disk group (minus any virtual disks). To remove the standalone disk group, set this parameter to `TRUE`. To keep standalone disk groups intact, set this parameter to `FALSE`.
`schedule`	Deletes only the schedule.

Delete Virtual Disk On A Disk Pool

Description

This command deletes either normal or thin virtual disks on a disk pool.

CAUTION: All of the data in the virtual disk is lost as soon as you run this command.

Syntax

```
delete (allVirtualDisks |
virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ... virtualDiskNameN] | allVirtual disks)
[removeDiskGroup=(TRUE | FALSE)
retainRepositoryMembers=(TRUE | FALSE)]
force=(TRUE | FALSE)
schedule
retainRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`allVirtualDisks`	This parameter deletes all of the virtual disks on a disk pool.
`virtualDisk` or `virtualDisks`	The name of the virtual disk that you want to delete. You can enter more than one virtual disk name. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").
`removeDiskGroup`	Deleting the last virtual disk in a disk group does not delete the disk group. You can have a standalone disk group (minus any virtual disks). To remove the standalone disk group, set this parameter to `TRUE`. To keep standalone disk groups intact, set this parameter to `FALSE`.
`force`	Use this parameter to force the immediate deletion of a virtual disk even if the RAID controller modules are performing other operations. To immediately force the deletion of a virtual disk, set this parameter to `TRUE`. To wait until the RAID controller modules have finished performing other operations, do not use this parameter or set this parameter to `FALSE`.
`schedule`	
`retainRepositoryMembers`	When you delete a thin virtual disk, the associated repository virtual disk is deleted by default. However, when the `retainRepositoryMembers` is set to `TRUE`, the repository virtual disk is retained. For normal virtual disks, this parameter has no effect.

Notes

When you use the `allVirtualDisks` parameter, this command deletes virtual disks until all of the virtual disks are removed or until an error is encountered. If an error is encountered, this command does not try to delete the remaining virtual disks. Deleting virtual disks from different disk groups is possible. All of the disk groups that become empty are deleted if you set the `removeDiskGroup` parameter to `TRUE`.

Diagnose RAID Controller Module

Description

This command runs diagnostic tests on the RAID controller module.

The `testID` parameter takes the following options, which you can use to verify that a RAID controller module is functioning correctly:

- **1** — Reads the test
- **2** — Performs a data loop-back test
- **3** — Writes the test

The diagnostic tests consist of loop-back tests in which data is written to physical disks and read from the physical disks.

Syntax

```
diagnose controller [(0 | 1)]
loopbackPhysicalDiskChannel=(allchannels | (1 | 2))
testID=(1 | 2 | 3)
[patternFile="filename"]
```

Parameters

Parameter	Description
`controller`	RAID controller module on which to run the diagnostic tests. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier. If you do not specify a RAID controller module, the storage management software returns a syntax error.
`loopbackPhysicalDiskChannel`	Physical disk channel on which to run the diagnostic tests. You can choose to run the diagnostics on all channels or select a specific channel on which to run diagnostics. Valid physical disk channel values are `1` or `2`. **NOTE:** When you run a data loopback test, you can optionally specify a file that contains a data pattern. If you do not specify a file, the firmware provides a default pattern.
`testID`	Identifier for the diagnostic test to run. The identifier and corresponding tests are: • **1** — Reads the test • **2** — Write test • **3** — Data loop-back test

Parameter	Description
	NOTE: When you run a data loop-back test, you can optionally specify a file that contains a data pattern. If you do not specify a file, the RAID controller module firmware provides a default pattern.
`patternFile`	File path and file name of the file that contains a data pattern to use as test data. Enclose the path and file name with quotation marks (" ").

Diagnose RAID Controller Module iSCSI Host Cable

Description

NOTE: This command valid only for the PowerVault MD3800i storage array.

This command runs diagnostic tests on the copper cables between the iSCSI Host interface cards and a RAID controller module. You can run diagnostics on a selected port or all ports. The ports must be able to support the cable diagnostics. If the ports do not support cable diagnostics an error is returned

Syntax

```
diagnose controller [(0 | 1)]
iscsiHostPort[(all | (0 | 1))]
testID=cableDiagnostics
```

Parameters

Parameter	Description
`controller`	Controller specifies the RAID controller module on which you want to run the cable diagnostic test. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. Enclose the RAID controller module identifier in square brackets ([ ]). If a RAID controller module is not specified, the MD storage management software returns a syntax error.
`iscsiHostPort`	Specifies the iSCSI host port on which you want to run the diagnostic tests. You can either run the diagnostics on all iSCSI host ports or run it on a specific iSCSI host. Valid values for the specific iSCSI host port identifiers are `0` or `1`. Enclose the iSCSI host port identifier in square brackets ([ ]).

Parameter	Description
`testID`	Specifies the identifier for the diagnostic test you want to run. For this diagnostic test, the value is `cableDiagnostics`.

Notes

When you run the cable diagnostic test, the firmware returns the following information:

- Port: The port on which the diagnostic test was run.
- The date and time the test was run.
- RAID Controller Module
- Status:
  - OK: All of the pairs of cables are good and do not have any faults.
  - Open: One or more of the four pairs of cables are open.
  - Short: One or more of the four pairs of cables are shorted.
- Incomplete: One or more of the four pairs returned incomplete or invalid test results.
- Cable Length: The length of the cables are listed in meters and the following information about the cables is returned:
  - When the cable status is **OK**, the approximate lengths of the cable pairs are returned. The lengths of the cable pairs are shown as a range (L1-L2), which are the shortest and the longest lengths of the cable pairs.
  - If the cable status is **Open** or **Short**, the approximate distance to the failure in the cable pairs. If there is one failure, the length is reported for that cable pair. If there is more than one failure, the information returned is both the shortest and longest lengths to the failures. The lengths are listed as a range (L1-L2) where L1<L2.
  - If the cable status is **Incomplete**, the information returned are the lengths for the shortest and longest cable pairs that the firmware can successfully test. The lengths are listed for the valid cable pairs as a range (L1-L2) where L1<L2.
- Raw Register values for the cable diagnostic registers. The values are in a hexadecimal format:
  - Two bytes show the combined cable status (four bits per port).
  - Four two-byte numbers show the length of each channel.

Diagnose Remote Replication

Description

NOTE: This command valid only with Fibre Channel storage arrays.

This command tests the connection between the specified primary virtual disks and the replication virtual disks on a storage array with the Remote Replication premium feature enabled.

Syntax

```
diagnose remoteReplication (primary
[primaryVirtualDiskName] |
primaries [primaryVirtualDiskName1 ...
primaryVirtualDiskNameN])
testID=connectivity
```

Parameter

Parameter	Description
`primary` or `primaries`	The name of the primary virtual disk of the remote replication pair to be tested. More than one primary virtual disk name can be entered. Enclose the primary virtual disk names in square brackets ([ ]). If the primary virtual disk name has special characters, the primary virtual disk name must also be enclosed in double quotation marks (" ").

Disable Storage Array Feature

Description

This command disables a storage array feature. Run the `show storageArray` command to display a list of the feature identifiers for all enabled premium features in the storage array.

Syntax

```
disable storageArray (featurePack | feature=featureAttributeList)
```

The *featureAttributeList* can be one or more of these attribute values. If you enter more than one attribute value, separate the values with a space.

- virtualDiskCopy
- snapshot
- asyncReplication
- remoteReplication
- mixedPhysicalDiskTypes
- goldKey
- driveSecurity
- enterpriseSecurityKeyMgr
- thinProvisioning
- storagePartition[2|4|8|16|32|64|96|128|192|256|512|Max]

The form for this attribute value is a combination of the alphabetical term merged with the numerical value, for example, storagePartition256

- physicalDiskSlotLimit[16|24|32|48|60|64|72|96|112|10|128|136|144|180|192|256| 272|300|360|384|448| 480|Max]

  The form for this attribute value is a combination of the alphabetical term merged with the numerical value, for example, physicalDiskSlotLimit360
- ssdCache
- SSDSupport
- highPerformanceTier
- raid6

NOTE: The following attribute is for enabling the snapshot (legacy) feature only.

- snapshot[2|4|8|16]

The form for this attribute value is a combination of the alphabetical term merged with the numerical value, for example, snapshot16

NOTE: The remoteReplication attribute is valid only for firmware versions before 7.84. Starting with firmware 7.84 the remoteReplication attribute is replaced by remoteReplication.

- remoteReplication[8|16|32|64|128]

The form for this attribute value is a combination of the alphabetical term merged with the numerical value, for example, remoteReplication128

Parameters

None.

Additional Information

If you specify the `remoteReplication` parameter, this command disables the Remote Replication premium feature and takes away the structure of the replication repository virtual disk.

If you disable the High Performance Tier feature, all of the included features are disabled.

To use the High Performance Tier premium feature, you must configure a storage array as:

- `SHIPPED_ENABLED`
- `SHIPPED_ENABLED=FALSE; KEY_ENABLED=TRUE`

Display Automatic Support Bundle Collection Configuration

Description

This command displays the automatic support bundle collection settings.

NOTE: This command must be run from the command line. It cannot be run in the MD Storage Manager script editor.

Syntax

```
supportBundle auto show
```

Parameters

None.

Display Support Bundle Collection Schedule

Description

This command displays the schedule for collecting support bundles for all storage arrays.

NOTE: This command must be run from the command line. It cannot be run in the MD Storage Manager script editor.

Syntax

```
supportBundle schedule show
```

Parameters

None.

Download Enclosure Management Module Firmware

Description

This command downloads firmware for the enclosure management module.

Syntax

```
download (allEnclosures | enclosure [enclosureID])
firmware file="filename"
```

Parameters

Parameter	Description
`enclosure`	Identifies the enclosure to which to load new firmware. Enclosure ID values are `0` to `99`. You must put brackets ([ ]) around the enclosure ID value. You can use the following parameters: • The `allEnclosures` parameter, which downloads new firmware to all of the EMMs in the storage array • The `enclosure` parameter, which downloads new firmware to a specific EMM. If you need to download new firmware to more than one EMM, but not all RAID controller modules, you must enter this command for each storage array.
`file`	File path and file name of the file that contains the firmware image. You must put quotation marks (" ") around the firmware image file path and file name.

Download Physical Disk Firmware

Description

This command downloads a firmware image to a physical disk.

CAUTION: Downloading physical disk firmware incorrectly can result in damage to the physical disks or loss of data.

Before attempting to download physical disk firmware, you must take the following precautions:

1. Stop all I/O activity to the storage array before downloading the firmware image.
2. Ensure the firmware image file is compatible with the physical disk enclosure. If you download a file that is not compatible with the selected physical disk enclosure, the enclosure might become unusable.
3. Do not make any configuration changes to the storage array while downloading the physical disk firmware. Attempting to make a configuration change can cause the firmware download to fail and make the selected physical disks unusable.

You can use this command to test the firmware on one physical disk before installing the firmware on all of the physical disks in a storage array. (Use the `Download Storage Array Physical Disk Firmware` command to download firmware on all of the physical disks in the storage array. See Download Storage Array Physical Disk Firmware.) This command blocks all I/O activity until the download finishes or fails. The download returns one of the following statuses:

- Successful
- Unsuccessful With Reason
- Never Attempted With Reason.

This command is used to download a firmware image to only one physical disk at a time. If you use this command in a script, ensure that you use this command only once. If you use this command more than once, the operation may fail. You can download firmware images to all of the physical disks in a storage array at one time by using the `Download Storage Array Physical Disk Firmware` command.

Syntax

```
download physicalDisk [enclosureID,slotID]
firmware file="filename"
```

Parameters

Parameter	Description
`physicalDisk`	Physical disk to which to download the firmware image. Specify the enclosure ID and slot ID for the physical disk. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put brackets ([ ]) around the enclosure ID value and slot ID value.
`filename`	File path and file name of the file that contains the firmware image. You must put quotation marks (" ") around the firmware image file path and file name.

Download Storage Array Firmware/NVSRAM

Description

This command downloads firmware and, optionally, NVSRAM values for the RAID controller module in the storage array. If you want to download only NVSRAM values, use the `downLoad storageArray NVSRAM` command.

Syntax

```
download storageArray firmware [, NVSRAM]
file="filename" [, "NVSRAM-filename"]
[downgrade=(TRUE | FALSE)]
[activateNow=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`NVSRAM`	Specifies that you want to download a file with NVSRAM values when you download a firmware file. You must not put brackets around this parameter. Include a comma after the term **firmware**.
`file`	File path and name of the file that contains the firmware. Valid file names must end with a **.dlp** extension. You must put quotation marks (" ") around the file name.
`NVSRAM-filename`	File path and name of the file that contains the NVSRAM values. Valid file names must end with a **.dlp** extension. You must put quotation marks (" ") around the NVSRAM file name. You must include a comma after the firmware file name.
`downgrade`	Specifies that you are loading firmware that is a previous version. The default value is `FALSE`. Set downgrade to `TRUE` if you want to download an earlier version of firmware.
`activateNow`	Activates the firmware and NVSRAM images. The default value is `TRUE`. If you set `activateNow` to `FALSE`, you must use the `activate storageArray firmware` command to activate the firmware and NVSRAM values at a later time.

Download Storage Array Physical Disk Firmware

Description

This command downloads firmware images to all of the physical disks in the storage array.

Syntax

```
download storageArray physicalDiskFirmware file= ”filename”
[file=”filename2”... file=”filenameN”]
```

Parameter

Parameter	Description
`file`	File path and name of the file that contains the firmware. Valid file names must end with a **.dlp** extension. You must put quotation marks (" ") around the file name.

Additional Information

When you run this command, you can download more than one firmware image file to the physical disks in a storage array. The number of firmware image files that you can download depends on the storage array. The MD storage management software returns an error if you try to download more firmware image files than the storage array can accept.

You can schedule downloads for multiple physical disks at the same time, including multiple physical disks in a redundant disk group. Each firmware image file contains information about the physical disk types on which the firmware image runs. The specified firmware images can be downloaded only to a compatible physical disk. Use the `download physicalDisk firmware` command to download a firmware image to a specific physical disk.

The `download storageArray physicalDiskFirmware` command blocks all I/ O activity until a download try has been made for each candidate physical disk or you run the `stop storageArray downloadPhysicalDiskFirmware` command. When the `download storageArray physicalDiskFirmware` command finishes downloading the firmware image, each candidate physical disk shows the download status for each physical disk. One of these statuses is returned:

- Successful
- Unsuccessful With Reason
- Never Attempted With Reason

Download Storage Array NVSRAM

Description

This command downloads NVSRAM values for the storage array RAID controller module.

Syntax

```
download storageArray NVSRAM file="filename"
[healthCheckMelOverride=(TRUE | FALSE)]
```

Parameter

Parameter	Description
`file`	The file path and the file name that contains the NVSRAM values. Enclose the NVSRAM file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\dnld \afrm.dlp"` Valid file names have a `.dlp` extension.
`healthCheckMelOverride`	The setting to override the results of the health check of the major event log (MEL). MEL validation still occurs, it is not bypassed. If the MEL check fails, you can bypass the failure by using this parameter when running the command. Before any download, the RAID controller module checks the event log to determine if an event has occurred that might prevent the new NVSRAM from successfully downloading. If such an event has occurred, the RAID controller module normally does not download new NVSRAM. This parameter forces the RAID controller module to download new NVSRAM. The default value is **`FALSE`**. Set this value to **`TRUE`** if you want to force the RAID controller module to download new NVSRAM.

Enable Disk Group Security

Description

This command converts a non-secure disk group to a secure disk group.

Syntax

`enable diskGroup [`*`diskGroupName`*`] security`

Parameter

Parameter	Description
`diskGroup`	The alphanumeric identifier (including - and _) of the disk group that you want to place in the Security Enabled state. Enclose the disk group identifier in square brackets ([ ]).

Additional Information

To successfully run this command, the following conditions must be met:

- All physical disks in the disk group must be self-encrypting disk (SED) physical disks.
- The Physical Disk Security premium feature must be enabled.
- The storage array security key must be set.
- The disk group must be Optimal, and not have snapshot virtual disks or repository virtual disks.

The RAID controller module firmware creates a lock that restricts access to the SED physical disks. SED physical disks have a state called Security Capable. When you create a security key, the state is set to Security Enabled, which restricts access to all SED physical disks that exist within the storage array.

Enable Disk Pool Security

Description

This command converts a non-secure disk pool to a secure disk pool.

Syntax

```
enable diskPool [diskPoolName] security
```

Parameter

Parameter	Description
`diskPool`	The alphanumeric identifier (including - and _) of the disk pool that you want to place in the Security Enabled state. Enclose the disk pool identifier in square brackets ([ ]).

Notes

All of the physical disks that comprise the disk pool must be security capable.

Enable or Disable ODX

This command turns on or turns off Offloaded Data Transfer (ODX) for a storage array. The storage array comes with ODX turned on. Use this command to turn off ODX if you do not want to run ODX, or if ODX is causing issues with the storage array. The only storage array that is affected is the storage array on which the command is run.

Syntax

```
set storageArray odxEnabled=(TRUE | FALSE)
```

Parameter

Parameter	Description
`odxEnabled`	The setting to turn on or turn off ODX. To turn on ODX set this parameter to **TRUE**. To turn off ODX set this parameter to **FALSE**. The default setting is that ODX is turned on.

NOTE: ODX provides a way to transfer data without using buffered read and buffered write operations and does not require direct host involvement with the data transfer operation. When ODX is not enabled, data is read from the source storage to the host, and then written to the target storage from the host. With ODX enabled, the data transfer operations are directly managed by the storage infrastructure. The data is moved directly from the source storage to the target storage without going through the host.

Enable Or Disable SSD Cache For A Virtual Disk

Description

This command turns on or off caching using the SSD cache feature for a specific virtual disk. The virtual disk can be either a standard virtual disk, a snapshot virtual disk, or a consistency group snapshot virtual disk.

Syntax Applicable To A Standard Virtual Disk

```
set virtualDisk ["virtualDiskName"] ssdCacheEnabled=(TRUE | FALSE)
```

Syntax Applicable To A Snapshot Virtual Disk

```
set snapVirtualDisk ["snapVirtualDiskName"] ssdCacheEnabled=(TRUE | FALSE)
```

Syntax Applicable To A Consistency Group Snapshot Virtual Disk

```
set cgSnapVirtualDisk ["cgSnapVirtualDiskName"] ssdCacheEnabled=(TRUE | FALSE)
```

Parameters

Parameter	Description
`virtualDisk`, `snapVirtualDisk`, or `cgSnapVirtualDisk`	The name of the specific virtual disk for which you want to turn on or off the SSD cache. Enclose the virtual disk name in double quotation marks (" ") inside of square brackets ([ ]).
`ssdCacheEnabled`	To turn on SSD cache, set this parameter to `TRUE`. To turn off SSD cache, set this parameter to `FALSE`.

Notes

You can turn the SSD cache on or off for only one virtual disk at a time.

When you turn off SSD cache for a virtual disk, the SSD cache for that virtual disk is purged.

Enable or Disable VAAI

This command turns on or turns off VMware vStorage API Array Architecture (VAAI) for a storage array. The storage array comes with VAAI turned on. Use this command to turn off VAAI if you do not want to run VAAI, or if VAAI is causing issues with the storage array. The only storage array that is affected is the storage array on which the command is run.

Syntax

```
set storageArray vaaiEnabled=(TRUE | FALSE)
```

Parameter

Parameter	Description
`vaaiEnabled`	The setting to turn on or turn off VAAI. To turn on VAAI set this parameter to **TRUE**. To turn off VAAI set this parameter to **FALSE**. The default setting is that VAAI is turned on.

NOTE: VAAI enables some storage tasks to be offloaded from the server hardware to a storage array. Offloading the storage tasks to a storage array increases host performance by reducing the tasks the host must perform.

This command is not supported on the CE4900, CE7900, or the CE7922 storage arrays.

Enable RAID Controller Module Data Transfer

Description

This command revives a RAID controller module that quiesces while running diagnostics.

Syntax

```
enable controller [(0 | 1)] dataTransfer
```

Parameter

Parameter	Description
`controller`	RAID controller module that you want to revive. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed

Parameter	Description
	from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier. If you do not specify a RAID controller module, the storage management software returns a syntax error.

Enable Storage Array Feature

Description

This command enables a premium feature using a feature key file.

Syntax

```
enable storageArray (featurePack | feature) file="filename"
```

Parameter

Parameter	Description
`file`	File path and file name of a valid feature key file. Valid file names for feature key files must end with a **.key** extension. You must put quotation marks (" ") around the file path and file name.

Additional Information

A feature pack is a predefined set of premium features, such as Storage Partitioning and Remote Replication. These premium features are combined for the convenience of the users.

A premium feature is an additional application to enhance the capabilities of a storage array.

NOTE: Before you enable the High Performance Tier premium feature, stop all host I/O operations to the storage array. When you enable the High Performance Tier premium feature, both RAID controller modules in the storage array immediately reboot.

Establish Remote Replicated Pair

Description

Use this command to complete a remote replicated pair on the remote storage array by adding a secondary virtual disk to an existing remote replication group. Before you run this command, the remote replication group must exist and the primary virtual disk must exist in the remote replication group. After this command successfully completes, remote replication starts between the primary virtual disk and the secondary virtual disk.

The two virtual disks comprising a remote replicated pair function as a single entity. Establishing a remote replicated pair allows you to perform actions on the entire replicated pair versus the two individual virtual disks.

Syntax

```
establish asyncRemoteReplication virtualDisk="secondaryVirtualDiskName"
asyncRemoteReplicationGroup="asyncRemoteReplicationGroupName"
primaryVirtualDisk="primayVirtualDiskName"
```

Parameters

Parameter	Description
`asyncRemoteReplication`	The name of an existing virtual disk on the remote storage array that you want to use for the secondary virtual disk. Enclose the virtual disk name in double quotation marks (" ").
`asyncRemoteReplicationGroup`	The name of an existing remote replication group that you want to use to contain the remote replicated pair. Enclose the remote replication group name in double quotation marks (" ").
`primaryVirtualDisk`	The name of an existing virtual disk on the local storage array that you want to use for the primary virtual disk. Enclose the virtual disk name in double quotation marks (" ").

Notes

A remote replicated pair is comprised of two virtual disks, a primary virtual disk and a secondary virtual disk, that contain identical copies of the same data. The replicated pair is a part of a remote replication group, which allows the replicated pair to synchronize at the same time as any other replicated pairs within the remote replication group.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

When you choose the primary virtual disk and the secondary virtual disk, the secondary virtual disk must be of equal or greater size than the primary virtual disk. The RAID level of the secondary virtual disk does not have to be the same as the primary virtual disk.

Increase Virtual Disk Capacity On A Disk Pool

Description

This command defines the properties for a virtual disk. You can use most parameters to define properties for one or more virtual disks. You also can use some parameters to define properties for only one virtual disk. The syntax definitions are separated to show which parameters apply to several virtual disks and which apply to only one virtual disk. The syntax for virtual disk mapping is listed separately.

NOTE: In configurations where disk groups consist of more than 32 virtual disks, the operation can result in host I/O errors or internal RAID controller module reboots due to the expiration of the timeout period before the operation completes. If you experience host I/O errors or internal RAID controller module reboots, bring the host to a quiescent state I/O and try the operation again.

Syntax Applicable To One Or More Virtual Disks

```
start increaseVolCapacity virtualDisk = virtualDiskName incrementalCapacity =
incrementalCapacity [addDrives = (drive-spec-list)]
```

Parameter

Parameter	Description
`addPhysicalDisks`	The setting to add new physical disks to the virtual disk. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for the physical disk. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for the physical disk. Enclosure ID values are `0` to `99`. Drawer ID values are `0` to `4`. Slot ID values are `0` to `31`. Enclose the enclosure ID value, drawer ID value, and the slot ID value in parentheses. Use this parameter with the `addCapacity` parameter if you need to specify additional physical disks to accommodate the new size. Each disk pool must be created from a minimum of 11 physical disks specified. There is no upward limit of physical disks supported in a single disk pool, but the number cannot exceed the number of physical disks supported on the storage array.

Load Storage Array DBM Database

Description

This command uploads a Database Management (DBM) database image from a file or from cache. This command restores a storage array DBM database to the exact configuration that existed when the database image was captured using the `save storageArray dbmDatabase` command. The data in a file can be just RAID configuration data or all data, including data for RAID configuration, disk groups and disk pools. The data in a cache location always includes all data.

Before using this command with the `file` option, you must first obtain a validator string (a security code) from your Technical Support representative. To obtain a validator, use the `save storageArray dbmValidator` command to generate an XML file that contains validator information.

Syntax

```
load storageArray dbmDatabase
((file="filename" validator="validatorValue") | sourceLocation=onboard)
```

```
[controller [(0|1)]]
[contentType=(raidConfigration | all)]
```

Parameters

Parameter	Description
`file`	The file path and the file name of the DBM database you want to upload. Enclose the file name in double quotation marks (" "). For example: file="C:\Array Backups \DBMbackup_03302010.dbm". This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.
`validator`	The alphanumeric security code required to restore a storage array to an existing configuration. Use the `save storageArray dbmValidator` command to generate the required validation information XML file.
`sourceLocation`	This parameter specifies the location from which to retrieve backup database information. The parameter must be included for consistency, but the only allowed value is `onboard`.
`controller`	This parameter specifies the RAID controller module from which data will be exclusively retrieved, if `sourceLocation` is set to `cache`. Enclose the RAID controller module identifier in square brackets ([ ]). If the `controller` parameter is not specified, data might be retrieved from either RAID controller module. Valid RAID controller module identifiers are `0` or `1`, where `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1.
`contentType`	This parameter specifies the content type of the data that is retrieved. • If the parameter is set to `raidConfiguration`, the RAID configuration data is retrieved. The RAID configuration data includes disk groups and associated virtual disks. • If the parameter is set to `all`, all of the data including disk pool configuration data is retrieved. The default is `raidConfiguration`.

Notes

Depending on the size of the database image, restoring the database might take up to 30 minutes. The host software does not show the RAID controller modules in an optimal state till all actions for loading the database image are completed on the controllers.

Modify Disk Pool

Description

This command adds capacity to a disk pool or changes the RAID controller module ownership for the entire disk pool. These two operations are mutually exclusive.

Syntax

```
set diskPool [diskPoolName]
(addPhysicalDisks=[enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn] | addCapacity=(diskPoolCapacity)) [owner=(0 |
1)]
```

Parameters

Parameter	Description
`addDrives`	Identifies multiple physical disks to add to the disk pool. Each disk pool must be created from a minimum of 11 physical disks specified. There is no upward limit of physical disks supported in a single disk pool, but the number cannot exceed the number of physical disks supported on the storage array.
`addCapacity`	Use the `addCapacity` parameter to specify the amount of storage and automatically select physical disks to meet that capacity.
`owner`	The RAID controller module that owns the disk pool. Valid RAID controller module identifiers are `0` or `1`, where `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1. If you do not specify an owner, the RAID controller module firmware determines the owner.

Recopy Virtual Disk Copy

Description

This command reinitiates a virtual disk copy operation by using an existing virtual disk copy pair.

NOTE: See Preparing Host Servers To Recopy A Virtual Disk.

CAUTION: Before you create a new copy of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk and (if applicable, the target disk) to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk(s) in Windows or unmounting the virtual drive in Linux helps to guarantee a stable copy of the drive for the virtual copy.

Syntax

```
recopy virtualDiskCopy target [targetName]
[source [sourceName]]
[copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)]
copyType=(online | offline)
```

Parameters

Parameter	Description
`target`	Name of the target virtual disk for which to reinitiate a virtual disk copy operation. You must put brackets ([ ]) around the target virtual disk name. If the target virtual disk name has special characters, you must also put quotation marks (" ") around the target virtual disk name.
`source`	Name of the source virtual disk for which to reinitiate a virtual disk copy operation. You must put brackets ([ ]) around the source virtual disk name. If the source virtual disk name has special characters, you must also put quotation marks (" ") around the source virtual disk name.
`copyPriority`	Specifies the priority that the virtual disk copy has relative to host I/O activity. Valid entries are `highest`, `high`, `medium`, `low`, or `lowest`. NOTE: `CopyPriority` defines the amount of system resources used to copy the data between the source and target virtual disks of a virtual disk copy pair. If you select the highest priority level, the virtual disk copy uses the most system resources to perform virtual disk copy, which decreases performance for host data transfers.
`targetReadOnlyEnabled`	Specifies whether the target virtual disk is write enabled or read only. To be able to write to the target virtual disk, set this parameter to `FALSE`. To prevent writing to the target virtual disk, set this parameter to `TRUE`.

Parameter	Description
`copyType`	Creates a virtual disk copy with or a snapshot which enables you to continue to write to the source virtual disk while creating the virtual disk copy. To reinitiate a virtual disk copy with a snapshot, set this parameter to `online`. To reinitiate a virtual disk copy without a snapshot, set this parameter to `offline`.

Recover RAID Virtual Disk

This command creates a RAID virtual disk with the given properties without initializing any of the user data areas on the physical disks. Parameter values are derived from the Recovery Profile data file (recovery-profile.csv) for the storage array. You can create the recover virtual disk in an existing disk group or create a new disk group by using this command.

NOTE: You can run this command only from a command line. You cannot run this command from the GUI script editor. You cannot use the storage managment GUI to recover a virtual disk.

Syntax

```
recover virtualDisk (physicalDisk=(enclosureID,drawerID,slotID) |
physicalDisks=(enclosureID1,drawerID1,slotID1
... enclosureIDn,drawerIDn,slotIDn) |
diskGroup=diskGroupName))
[newDiskGroup=diskGroupName]
userLabel="virtualDiskName" virtualDiskWWN="virtualDiskWWN"
capacity=virtualDiskCapacity
offset=offsetValue
raidLevel=(0 | 1 | 5 | 6)
segmentSize=segmentSizeValue
dssPreallocate=(TRUE | FALSE)
SSID=subsystemVirtualDiskID
[owner=(0 | 1)]
[cacheReadPrefetch=(TRUE | FALSE)]
[allocateReservedSpace=(TRUE | FALSE)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
physicalDisk or physicalDisks	The physical disks that you want to assign to the disk group that will contain the virtual disk that you want to recover. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each physical disk that you assign to the virtual disk. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each physical disk that you assign to the virtual disk. Enclosure ID values are **0** to **99**. Drawer ID values are **0** to **4**. Slot ID values are **0** to **31**. Enclose the

	enclosure ID values, the drawer ID values, and the slot ID values insqurare brackets ([ ]).
diskGroup	The name of an existing disk group in which you want to create the virtual disk. (To determine the names of the disk groups in your storage array, run the **show storageArray profile** command.)
newDiskGroup	The name that you want to give a new disk group. Enclose the new disk group name in double quotation marks (" ").
userLabel	The name of the virtual disk that you want to recover. The user label has two parts: • The virtual disk name. Enclose the virtual disk name in double quotation marks (" "). • The world wide name of the virtual disk, virtualDiskWWN, in the form of a 16 byte identifier, for example, 60080E500017B4320000000049887D77. Enclose the identifier in double quotation marks (" "). You must enter both the virtual disk name and the world wide name of the virtual disk. You must enter the virtual disk name first. For example: userLabel="engdata" virtualDiskWWN=60080E500017B4320000000049887D77
capacity	The size of the virtual disk that you are adding to the storage array. Size is defined in units of **bytes**, **KB**, **MB**, **GB**, or **TB**.
offset	The number of blocks from the start of the disk group to the start of the referenced virtual disk.
raidLevel	The RAID level of the disk group that contains the physical disks. Valid values are **0**, **1**, **5**, or **6**.
segmentSize	The amount of data (in KB) that the RAID controller module writes on a single physical disk in a disk group before writing data on the next physical disk. Valid values are **8**, **16**, **32**, **64**, **128**, **256**, or **512**.
dssPreallocate	The setting to turn on or turn off allocating virtual disk storage capacity for future segment size changes. To turn on allocation, set this parameter to **TRUE**. To turn off allocation, set this parameter to **FALSE**.
SSID	The storage array subsystem identifier of a virtual disk. Use the **show virtual disk** command to determine the storage array subsystem identifier.

owner	The RAID controller module that owns the virtual disk. Valid RAID controller module identifiers are **0** or **1**, where **0** is the RAID controller module in slot 0, and **1** is the RAID controller module in slot 1. If you do not specify an owner, the RAID controller module firmware determines the owner.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to **FALSE**. To turn on cache read prefetch, set this parameter to **TRUE**.
dataAssurance	The setting to specify that a disk group, and the virtual disks within the disk group, has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk group. These settings are valid: • **none** – The disk group does not have data assurance protection. • **enabled** – The disk group has data assurance protection. The disk group supports protected information and is formatted with protection information enabled.
allocateReservedSpace	The setting to disable or enable the reserve capacity on the physicalDisks in a disk group. The controller firmware automatically reserves additional capacity on each drive when creating a disk group. To prevent the controller firmware from reserving additional capacity on each physicalDisk in a disk group, set this parameter to **FALSE**. The default value is **TRUE**.

NOTE:

The MD storage management software collects recovery profiles of the monitored storage arrays and saves the profiles on a storage management station.

The physicalDisk parameter supports both high-capacity expansion enclosures and low-capacity expansion enclosures. A high-capacity expansion enclosure has drawers that hold the physical disks. The drawers slide out of the expansion enclosure to provide access to the physical disks. A low-capacity expansion enclosure does not have drawers. For a high-capacity expansion enclosure, you must specify the identifier (ID) of the expansion enclosure, the ID of the drawer, and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, you need only specify the ID of the expansion enclosure and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, an alternative method for identifying a location for a physical disk is to specify the ID of the expansion enclosure, set the ID of the drawer to **0**, and specify the ID of the slot in which a physical disk resides.

If you attempt to recover a virtual disk using the physicalDisk parameter or the physicalDisks parameter and the physical disks are in an unassigned state, the RAID controller module automatically creates a new disk group. Use the newVirtualDiskGroup parameter to specify a name for the new disk group.

You can use any combination of alphanumeric characters,underscore (_), hyphen(-), and pound (#) for the names. Names can have a maximum of 30 characters.

The owner parameter defines which RAID controller module owns the virtual disk. The preferred RAID controller module ownership of a virtual disk is the RAID controller module that currently owns the disk group.

Preallocating Storage Capacity

The dssPreallocate parameter enables you to assign capacity in a virtual disk for storing information that is used to rebuild a virtual disk. When you set the dssPreallocate parameter to **TRUE**, the storage space allocation logic in the RAID controller module firmware preallocates the space in a virtual disk for future segment size changes. The preallocated space is the maximum allowable segment size. The dssPreallocate parameter is necessary for properly recovering virtual disk configurations that are not retievable from the RAID controller module data base. To turn off the preallocation capability, set dssPreallocate to **FALSE**.

Segment Size

The size of a segment determines how many data blocks that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8-KB segment holds 16 data blocks. A 64-KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the RAID controller module at run time. If the value that you entered is not valid, the RAID controller module returns a list of valid values. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests.

If the virtual disk is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of physical disks in the disk group that are used for data transfers.) In this case, multiple physical disks are used for the same request, but each physical disk is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of physical disks that are required to satisfy a data transfer request.

Cache Read Prefetch

Cache read prefetch lets the RAID controller module copy additional data blocks into cache while the RAID controller module reads and copies data blocks that are requested by the host from disk into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the RAID controller module reads into cache. Valid values for the cacheReadPrefetch parameter are **TRUE** or **FALSE**.

Reserved Space on Physical Disks

The controller firmware reserves 5 GB of space on each drive in a disk group or disk pool for future expansion of the DACstore. The space must be adjacent to the DACstore or it will not be reserved. The reserved space is defined only when you create the disk group or disk pool. Unassigned physical disks and hot spare physical disks will not have reserved space. After the disk group or disk pool has been created, the reserved space is persistent. If a storage array is being upgraded to controller firmware 8.20, disk groups or disk pools that were created under a previous version of controller firmware will not be affected; that is, the additional space will not be reserved on existing disk groups or disk pools.

Re-Create Remote Replication Repository Virtual Disk

Description

NOTE: This command is valid only with Fibre Channel storage arrays.

This command creates a new Remote Replication repository virtual disk (also called a replication repository virtual disk) by using the parameters defined for a previous replication repository virtual disk. The underlying requirement is that you have previously created a replication repository virtual disk. When using this command, the replication repository virtual disk can be defined in one of three ways:

- User-defined physical disks
- User-defined disk group
- User-defined number of physical disks

If you choose to define a number of physical disks, the RAID controller module firmware chooses which physical disks to use for the replication repository virtual disk.

Syntax

User-Defined Physical Disks

```
recreate storageArray replicationRepository repositoryRAIDLevel=(1 | 5 | 6)
repositoryPhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
```

```
enclosureIDN,drawerIDN,slotIDN) [enclosureLossProtect=(TRUE | FALSE)]
[dataAssurance=(none | enabled)]
```

User-Defined Disk Group

```
recreate storageArray replicationRepository
repositoryDiskGroup=diskGroupName
[freeCapacityArea=freeCapacityIndexNumber]
```

User-Defined Number of Drives

```
recreate storageArray replicationRepository
repositoryRAIDLevel=(1 | 5 | 6)
repositoryPhysicalDiskCount=numberOfPhysicalDisks [physicalDiskType=(SAS)]
[enclosureLossProtect=(TRUE | FALSE)]
[dataAssurance=(none | enabled)]
```

Parameters

Parameter	Description
`repositoryRAIDLevel`	The RAID level for the replication repository virtual disk. Valid values are `1`, `5`, or `6`.
`repositoryPhysicalDisks`	The physical disks for the replication repository virtual disk. Specify the enclosure ID and slot ID for each physical disk that you assign to the replication repository virtual disk. Enclosure ID values are `0` to `99`. Slot ID values are `1` to `32`. Enclose the enclosure ID values and the slot ID values in parentheses.
`repositoryDiskGroup`	The name of the disk group where the replication repository virtual disk is located.
`freeCapacityArea`	The index number of the free space in an existing disk group that you want to use to re-create the replication repository virtual disk. Free capacity is defined as the free capacity between existing virtual disks in a disk group. For example, a disk group might have the following areas: virtual disk 1, free capacity, virtual disk 2, free capacity, virtual disk 3, free capacity. To use the free capacity following virtual disk 2, you would specify: `freeCapacityArea=2` Run the `show diskGroup` command to determine if a free capacity area exists.
`repositoryPhysicalDiskCount`	The number of unassigned physical disks that you want to use for the replication repository virtual disk.
`enclosureLossProtect`	The setting to enforce enclosure loss protection when you create the replication repository virtual

Parameter	Description
	disk. To enforce enclosure loss protection, set this parameter to `TRUE`. The default value is `FALSE`.
physicalDiskType	The type of physical disk that you want to use for the replication repository virtual disk. You cannot mix physical disk types. You must use this parameter when you have more than one type of physical disk in your storage array. Valid physical disk types are : • SAS If you do not specify a physical disk type, the command defaults to SAS.
dataAssurance	The setting to specify that a disk pool has data assurance protection to make sure that the data maintains its integrity. When you use this parameter, only protected physical disks can be used for the disk pool. These settings are valid: • none – The disk pool does not have data assurance protection. • enabled – The disk pool has data assurance protection. The disk pool supports protected information and is formatted with protection information enabled. The default value is none.

Notes

If a value is entered for the repository storage space that is too small for the replication repository virtual disks, the RAID controller module firmware returns an error message that indicates the amount of space needed for the replication repository virtual disks. The command does not try to change the replication repository virtual disk. The command can be re-entered using the value from the error message for the repository storage space value.

When you assign the physical disks, if you set the `enclosureLossProtect` parameter to `TRUE` and have selected more than one physical disk from any one enclosure, the storage array returns an error. If you set the `enclosureLossProtect` parameter to `FALSE`, the storage array performs operations, but the disk group that you create might not have enclosure loss protection.

When the RAID controller module firmware assigns the physical disks, if you set the `enclosureLossProtect` parameter to `TRUE`, the storage array returns an error if the RAID controller module firmware cannot provide physical disks that result in the new replication repository virtual disk having enclosure loss protection. If you set the `enclosureLossProtect` parameter to `FALSE`, the storage array performs the operation even if it means that the replication repository virtual disk might not have enclosure loss protection.

Re-Create Snapshot

Description

This command starts a new copy-on-write operation by using an existing snapshot virtual disk. You can re-create a single snapshot virtual disk or recreate multiple virtual disks.

CAUTION: Before you create a new point-in-time image of a source virtual disk, stop any data access (I/O) activity or suspend data transfer to the source virtual disk to ensure that you capture an accurate point-in-time image of the source virtual disk. Close all applications, including Windows Internet Explorer, to make sure all I/O activity has stopped.

NOTE: Removing the drive letter of the associated virtual disk in Windows or unmounting the virtual drive in Linux helps guarantee a stable copy of the drive for the Snapshot.

Syntax

```
recreate snapshot (virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ...
virtualDiskNameN])
[userLabel="snapshotVirtualDiskName"
warningThresholdPercent=percentValue
repositoryFullPolicy (failSourceWrites |
failSnapshot)]
```

Parameters

Parameter	Description
`virtualDisk` or `virtualDisks`	Name of the specific virtual disk for which to start a fresh copy-on-write operation. You can enter more than one virtual disk name. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has special characters, you must also put quotation marks (" ") around the virtual disk name.
`userLabel`	Name of the snapshot virtual disk. You must put quotation marks (" ") around the snapshot virtual disk name. If you enter more than one snapshot virtual disk name, this command fails.
`warningThresholdPercent`	Percentage of repository capacity at which you receive a warning that the repository is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is 50 percent. NOTE: If `warningThresholdPercent` is not specified, the previously set value is used.
`repositoryFullPolicy`	Specifies how snapshot processing continues if the repository is full. You can choose to fail writes to the source virtual disk (`failSourceWrites`) or fail

Parameter	Description
	writes to the snapshot virtual disk (`failSnapShot`). The default value is `failSnapShot`. NOTE: If `repositoryFullPolicy` is not specified, the previously set value is used.

Additional Information

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

You can recreate a single snapshot virtual disk or multiple snapshot virtual disks with optional parameters:

- If you specify one or more of the optional parameters, the re-create operation processes each snapshot virtual disk separately.
- If you try to use the same user label for more than one virtual disk, the command fails.
- If you do not set the `warningThresholdPercent` parameter or the `repositoryFullPolicy` parameter, values that you previously set are used.

You can recreate multiple snapshot legacy virtual disks without optional parameters:

- If you list multiple snapshot legacy virtual disks to be re-created but do not specify any of the optional parameters, the re-create operation processes the snapshot virtual disks as a batch process.
- Validation checks for the necessary snapshot-restarted preconditions are performed before restarting any snapshot. If any of the listed snapshot virtual disks fail the validation, the entire command fails and the snapshot virtual disks are not re-created. If the validation is successful for all of the snapshot virtual disks in the list, but one or more of the snapshots in the list fails to restart, the entire command fails and none of the snapshots are re-created.
- During snapshot re-creation, all affected virtual disks (snapshots, base, and repository) are appropriately quiesced and I/O operations are resumed to all affected virtual disks after all snapshots have been successfully re-created.

NOTE: If the `warningThresholdPercent` parameter or the `repositoryFullPolicy` parameter is not specified, the previously set values are used. If you use the optional parameters, the re-create snapshot processes each snapshot virtual disk separately. If the same userLabel (name) is specified for more than one virtual disk, the command fails. If no optional parameters are specified, the re-create Snapshot processes the specified snapshot virtual disks as a batch, provided a validation check of all of the virtual disks runs successfully. If successful, the snapshots start the re-creation process and all of the affected virtual disks (snapshot, source, and repository) are quiesced until the process is complete.

NOTE: If no optional parameters are specified, the `recreate snapshot` command processes the specified snapshot legacy virtual disks as a batch, provided a validation check of all of the virtual disks runs successfully. If successful, the snapshots start the re-creation process and all of the affected virtual disks (snapshot, source, and repository) are quiesced until the process is complete.

NOTE: The Microsoft Virtual Shadow Copy Service (VSS) provider allows the recreation of multiple snapshots simultaneously.

Reduce Disk Pool Capacity

Description

This command reduces the capacity of the disk pool by logically removing the physical disks from the pool. This command uses the Dynamic Capacity Reduction (DCR) feature, which enables you to remove selected physical disks from a disk pool and re-use the removed physical disks as needed.

The maximum number of physical disks that you can remove from a disk pool in a single DCR operation is 12. You cannot reduce the disk pool to below the minimum disk pool size.

Syntax

```
set diskPool [diskPoolName]
removeDrives=(enclosureID1,drawerID1,slotID1
... enclosureIDn,drawerIDn,slotIDn)
```

Parameters

Parameter	Description
`diskPool`	The name of the disk pool for which you want to reduce capacity. Enclose the disk pool name in square brackets ([ ]).
`removeDrives`	The physical disks that you want to remove from the disk pool. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each physical disk that you want to remove. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each physical disk that you want to remove. Enclosure ID values are **0** to **99**. Drawer ID values are **0** to **4**. Slot ID values are **0** to **31**. Enclose the enclosure ID values, the drawer ID values, and the slot ID values in square brackets ([ ]).

Notes

Each disk pool name must be unique. You can use any combination of alphanumeric characters, underscore (_), hyphen(-), and pound (#) for the user label. User labels can have a maximum of 30 characters.

You might want to remove selected physical disks from a disk pool if you need to create a separate group or remove unneeded capacity from the storage array. The removed physical disks become unassigned physical disks and the data that existed on them is redistributed across the remaining physical disks in the disk pool. The capacity of the disk pool is decreased by the capacity of the physical disks that you remove. Removing physical disks from a disk pool is always carried out as a background task and the

virtual disks remain fully accessible during this operation. The progress of the removal operation is reported as part of the long running operations status.

This command fails if there is not enough free capacity on the physical disks that remain in the disk pool to store the data that was on the physical disks you remove from the disk pool. Reserve capacity in the disk pool might be used to store data that was on the physical disks you remove from the disk pool. However, if the reserve capacity falls below the critical threshold, a critical event is written to the major event log.

The RAID controller module firmware determines how many physical disks you can remove. You can then select the specific physical disks to be removed. The RAID controller module firmware bases the count on the amount of free space needed to redistribute the data across the remaining physical disks without consuming any capacity reserved for reconstructions. If the free capacity of the disk pool is already less than the Reserved Reconstruction Physical Disk Amount, the RAID controller module does not allow a DCR operation to begin.

A DCR operation might result in the configured capacity exceeding one or both of the Pool Utilization Thresholds. If so, normal threshold alerts are issued.

Register SNMP Community

Description

This command creates a new Simple Network Management Protocol (SNMP) community, community name string, and registers the new community as a known entity to the SNMP agent. Any attempt to register a new community with the same name as that of an existing community is rejected and the firmware returns an error message.

Syntax

```
create snmpCommunity communityName="snmpCommunityName"
```

Parameters

Parameter	Description
communityName	The name of the SNMP community that you want to create. Enclose the SNMP community name in double quotation marks (" ").

Register SNMP Trap Destination

Description

This command creates a new Simple Network Management Protocol (SNMP) trap destination. A trap destination is the SNMP manager that you want to have receive the trap messages.

Syntax

```
create snmpTrapDestination trapReceiverIP=ipAddress
communityName="communityName"
[sendAuthenticationFailureTraps=(TRUE | FALSE)]
```

Parameters

Parameter	Description
trapReceiverIP	The IP address of the SNMP manager to which you want to send trap messages.
communityName	The name of the SNMP community for which you want to send trap messages.
sendAuthenticationFailureTraps	This parameter turns on or turns off sending authentication failure messages to an SNMP manager. To send authentication failure messages, set the parameter to **TRUE**. To prevent sending authentication failure messages, set the parameter to **FALSE**. The default is **FALSE**.

Reinitialize Thin Virtual Disk

Description

This command re-initializes a thin virtual disk.

NOTE: Formatting a virtual disk starts a long-running operation that cannot be stopped.

Syntax

```
start virtualDisk [virtualDiskName] initialize
[existingRepositoryLabel=existingRepositoryName]
[diskPool=diskPoolName capacity=capacityValue] [retainRepositoryMembers=(TRUE|
FALSE)]
```

Parameters

Parameter	Description
`virtualDisk`	The name of the virtual disk for which you are starting the formatting. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").
`existingRepositoryLabel`	Causes the repository virtual disk to be replaced by the candidate virtual disk specified. • The value specified is an existing repository virtual disk user label. The virtual disk specified

Parameter	Description
	must be an unused repository virtual disk with the name in the proper form. • If the newly specified repository virtual disk is on a different disk pool, the thin virtual disk will change ownership to that pool. • The old repository virtual disk will be deleted by default.
`diskPoolUserLabel`	Use this parameter to allocate a new repository virtual disk with the specified capacity.
`retainRepositoryMember`	If this parameter is set to `TRUE`, the old repository is retained. By default, the old repository is deleted. This parameter is ignored if the existing repository is reused.

Remove Incomplete Remote Replicated Pair From Remote Replication Group

Description

This command removes an orphaned replicated pair virtual disk on the storage array. An orphaned replicated pair virtual disk exists when a member virtual disk in a remote replication group has been removed on one side of the remote replication group (either the primary side or secondary side) but not on the other side.

Orphan replicated pair virtual disks are detected when inter-RAID controller module communication is restored and the two sides of the replication configuration reconcile replication parameters.

Use this command when the replicate relationship is successfully removed on the local or remote storage array, but cannot be removed on the corresponding storage array because of a communication problem.

Syntax

```
remove asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
incompleteReplication virtualDisk="virtualDiskName"
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group that contains the orphaned virtual disk that you want to remove. Enclose the remote replication group name in double quotation marks (" "). If the remote replication group name has special characters, you also must enclose the remote replication group

Parameter	Description
	name in double quotation marks (" ") inside square brackets.
incompleteAsyncRemoteReplication	The name of the orphaned virtual disk that you want to remove from the remote replication group. Enclose the virtual disk name in double quotation marks (" ").

Remove Member Virtual Disk From Consistency Group

Description

This command removes a member virtual disk from a an existing snapshot consistency group. Optionally, you can delete the repository virtual disk members from the consistency group.

Syntax

```
set consistencyGroup ["consistencyGroupName"]
removeCGMemberVirtualDisk=memberName
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
consistencyGroupName	The name of the consistency group from which you want to remove a member. Enclose the consistency group name in double quotation marks (" ") inside square brackets ([ ]).
memberName	The name of the member virtual disk that you want to remove.
deleteRepositoryMembers	Determines whether to remove all of the repository members from the consistency group.

Remove Physical Disks From SSD Cache

Description

This command decreases the capacity of the SSD cache by removing Solid State Disks (SSDs).

Syntax

```
set ssdCache [ssdCacheName]
removePhysicalDisks=(enclosureID1,drawerID1,slotID1 ...
enclosureIDn,drawerIDn,slotIDn)
```

Parameters

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache from which you want to remove SSDs. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
`removePhysicalDisks`	The physical disks that you want to remove from the SSD cache. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for each SSD that you want to remove. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for each SSD that you want to remove. Enclosure ID values are `0` to `99`. Drawer ID values are `0` to `4`. Slot ID values are `0` to `31`. Enclose the enclosure ID values, the drawer ID values, and the slot ID values in parentheses.

Notes

You cannot remove all of the SSDs from the SSD cache using this command; at least one SSD must remain in the SSD cache. If you want to completely remove the SSD cache, use the `delete ssdCache` command instead.

Remove Remote Replication

Description

NOTE: This command valid only with Fibre Channel storage arrays.

This command removes the replication relationship between the primary virtual disk and the secondary virtual disk in a remote replication pair.

Syntax

```
remove remoteReplication (localVirtualDisk [virtualDiskName] |
localVirtualDisks [virtualDiskName1... virtualDiskNameN])
```

Parameter

Parameter	Description
`localVirtualDisk or localVirtualDisks`	The name of the primary virtual disk (the virtual disk on the local storage array) that you want to remove. You can enter more than one virtual disk

Parameter	Description
	name. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").

Remove Virtual Disk Copy

Description

This command removes a virtual disk copy pair.

Syntax

```
remove virtualDiskCopy target [targetName]
[source [sourceName]]
copyType=(online | offline)
```

Parameters

Parameter	Description
`target`	Name of the target virtual disk to remove. You must put brackets ([ ]) around the target virtual disk name. If the target virtual disk name has special characters, you must also put quotation marks (" ") around the target virtual disk name.
`source`	Name of the source virtual disk to remove. You must put brackets ([ ]) around the source virtual disk name. If the source virtual disk name has special characters, you must also put quotation marks (" ") around the source virtual disk name.
`copyType`	Identifies a virtual disk copy with a snapshot. If the virtual disk copy has an online copy snapshot, set this parameter to `online`. If the virtual disk copy does not have a snapshot, set this parameter to `offline`.

Remove Virtual Disk From Remote Replication Group

Description

This command removes a member virtual disk from an existing remote replication group. Optionally, you can delete the repository virtual disk members from the remote replication group.

This command is valid only on the local storage array that contains the remote replication group whose member virtual disk that you want to remove.

Syntax

```
remove virtualDisk ["virtualDiskName"]
asyncRemoteReplicationGroup="asyncRemoteReplicationGroupName"
[deleteRepositoryMembers=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`virtualDisk`	The name of the specific virtual disk that you want to remove from the remote replication group. Enclose the virtual disk name in double quotation marks (" ") inside square brackets ([ ]).
`asyncRemoteReplicationGroup`	The name of the remote replication group that contains the member virtual disk that you want to remove. Enclose the remote replication group name in double quotation marks (" ").
`deleteRepositoryMembers`	Determines whether to remove all of the repository members from the remote replication group.

Remove Virtual Disk LUN Mapping

Description

This command removes the logical unit number (LUN) mapping.

Syntax

```
remove (allVirtualDisks | virtualDisk
["virtualDiskName"] |
virtualDisks ["virtualDiskName1" ...
"virtualDiskNamen"] | accessVirtualDisk)
lunMapping (host="hostName" | hostGroup=
"hostGroupName")
```

Parameters

Parameter	Description
`allVirtualDisks`	Removes the LUN mapping from all virtual disks.
`virtualDisk` or `virtualDisks`	Name of the specific virtual disk to remove from the LUN mapping. You can enter more than one virtual disk name. You must put quotation marks (" ") and brackets ([ ]) around the virtual disk name. The virtual disk name and quotation marks must be inside the brackets.
`accessVirtualDisk`	Removes the access virtual disk.

Parameter	Description
	CAUTION: The host agent uses the access virtual disks to communicate in-band with a storage array. If you remove an access virtual disk mapping for a storage array from a host that has an agent running on it, the storage management software is no longer able to manage the storage array through the in-band agent.
`host`	Name of the host to which the virtual disk is mapped. You must put quotation marks (" ") around the host name.
`hostGroup`	Name of the host group that contains the host to which the virtual disk is mapped. You must put quotation marks (" ") around the host group name.

NOTE: You must use the `host` and `hostGroup` parameters when specifying a nonaccess virtual disk or an access virtual disk. The script engine ignores the `host` or `hostGroup` parameters when you use the `allVirtualDisks` or `virtualDisks` parameters.

Rename A Snapshot Virtual Disk

Description

This command renames an existing snapshot virtual disk.

Syntax

```
set snapVirtualDisk ["snapVirtualDiskName"]
userLabel="snapImageVirtualDiskName"
```

Parameters

Parameter	Description
`snapVirtualDisk`	The name of the snapshot virtual disk that you want to rename. Enclose the snapshot virtual disk name in double quotation marks (" ") inside of square brackets ([ ]).
`userLabel`	A new name that you want to give to the snapshot virtual disk. Enclose the new snapshot virtual disk name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

Rename SSD Cache

Description

This command changes the name of the SSD cache.

Syntax

```
set ssdCache [old_ssdCacheName] userLabel="new_ssdCacheName"
```

Parameters

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache that you want to rename. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
userLabel	The new name for the SSD cache. Enclose the name in double quotation marks (" ").You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the identifier. Identifiers can have a maximum of 30 characters.

Repair Virtual Disk Consistency

Description

This command repairs the consistency errors on a virtual disk.

Syntax

```
repair virtualDisk [virtualDiskName] consistency
consistencyErrorFile="filename"
[verbose=(TRUE | FALSE)]
```

Parameters

Parameter	Description
virtualDisk	Name of the specific virtual disk for which to repair consistency. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has

Parameter	Description
	special characters, you must also put quotation marks (" ") around the virtual disk name.
consistencyErrorFile	Name of the file that contains the consistency error information you use to repair the errors. You must put quotation marks (" ") around the file name.
verbose	Captures progress details, such as percent complete, and shows the progress detail information as virtual disk consistency is being repaired. To capture progress details, set this parameter to TRUE. To prevent capturing progress details, set this parameter to FALSE.

Replace Physical Disk

Description

This command redefines the composition of a disk group by replacing a physical disk that is in the disk group with either:

- An unassigned physical disk
- A fully integrated hot spare physical disk

Syntax

```
replace physicalDisk ([enclosureID,slotID] |
<wwID>)replacementphysicalDisk=enclosureID,slotID
[copyphysicalDisk]
[failphysicalDisk]
```

Parameters

Parameter	Description
physicalDisk	Specifies the physical disks to assign to the virtual disk that you want to create. Specify the enclosure ID and slot ID for each physical disk that you assign to the virtual disk. Enclosure ID values are 0 to 99. Slot ID values are 0 to 31. You must put brackets ([ ]) around the enclosure ID value and the slot ID value.
physicalDisk <wwID>	The World Wide Identifier (WWID) of the physical disk that you want to replace. You can use the WWID instead of the virtual disk name to identify the virtual disk. Enclose the WWID in angle brackets (< >).

Parameter	Description
`replacementPhysicalDisk`	Specifies the physical disks to assign to the virtual disk that you want to create. Specify the enclosure ID and slot ID for each physical disk that you assign to the virtual disk. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`.
`copyPhysicalDisk`	This parameter copies the data from the physical disk that you are replacing to the replacement physical disk.
`failPhysicalDisk`	This parameter causes the physical disk that you are replacing to be failed after the data has been copied to the replacement physical disk. This parameter can be used only when the copyPhysicalDisk parameter is used.

Reset iSCSI IP Address

Description

This command resets the IP address for the remote storage array to re-establish connection with the local storage array. You can use this command to notify the local storage array that the iSCSI IP address of the remote storage array have changed and need to be updated.

When establishing an remote replication relationship with an iSCSI connection, both storage arrays store a record of the IP address of the remote storage array of the remote replication configuration. If the IP address of an iSCSI port changes, the remote storage array that is attempting to use that port encounters a communication error.

The storage array with the changed IP address sends a message to each remote storage array associated with the remote replication groups that are configured to replicate over an iSCSI connection. Storage arrays that receive this message automatically update their remote-target IP address.

If the storage array with the changed IP address is unable to send its inter-RAID controller module message to a remote storage array, the system sends you an alert of the connectivity issue. Use the `reset` command to re-establish connection with the local storage array.

Syntax

```
reset (remoteStorageArrayName="storageArrayName" |
remoteStorageArrayWwid=<wwID>)
iscsiIpAddress
```

Parameters

Parameter	Description
`remoteStorageArrayName`	The name for the remote storage array for which you are resetting the iSCSI IP address. Enclose the storage array name in double quotation marks (" ").
`remoteStorageArrayNameWwid`	The World Wide Identifier (WWID) of the storage array for which you are resetting the iSCSI IP address. You can use the WWID instead of the storage array name to identify the storage array. Enclose the WWID in angle brackets (< >).

Reset Physical Disk

This command power cycles a physical disk in a disk group or a disk pool to aid in the recovery of a physical disk that is exhibiting inconsistent or non-optimal behavior.

By power cycling a physical disk, some errors that cause inconsistent or non-optimal behavior can be cleared. This avoids replacing a physical disk when it is experiencing only a transient, non-fatal error, and the physical disk can remain operational. Resetting a physical disk in this way reduces disruptions and avoids replacing a physical disk.

If the problem cannot be corrected by power cycling the physical disk, the data is copied from the physical disk and the physical disk is powered down for replacement.

Syntax

```
reset physicalDisk([enclosureID,drawerID,slotID] | <"wwID">)
```

Parameters

Parameter	Description
`physicalDisk`	The location of the physical disk that you want to replace. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value of the physical disk that you want to revive. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value of the physical disk that you want to revive. Enclosure ID values are **0** to **99**. Drawer ID values are **0** to **4**. Slot ID values are **0** to **31**. Enclose the enclosure ID value, the drawer ID value, and the slot ID value in square brackets ([ ]).
`physicalDisk`	The World Wide Identifier (WWID) of the physical disk that you want to replace. Enclose the WWID in double quotation marks (" ") inside angle brackets (< >).

NOTE: The physicalDisk parameter supports both high-capacity expansion enclosures and low-capacity expansion enclosures. A high-capacity expansion enclosure has drawers that hold the physical disks. The drawers slide out of the expansion enclosure to provide access to the physical disks. A low-capacity expansion enclosure does not have drawers. For a high-capacity expansion enclosure, you must specify the identifier (ID) of the expansion enclosure, the ID of the drawer, and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, you need only specify the ID of the expansion enclosure and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, an alternative method for identifying a location for a physical disk is to specify the ID of the expansion enclosure, set the ID of the drawer to **0**, and specify the ID of the slot in which a physical disk resides.

After power cycling the physical disk, the controller must verify the physical disk to be functional before it can be placed back into use. If the physical disk cannot be verified it is marked as a failed physical disk. Power cycling a failed physical disk requires global hot spare physical disks and full copy back to a replacement physical disk after it is brought into service.

Failed physical disks cannot be reset by a power cycle more than once in a 24 hour period, and in some cases cannot be reset more than once. Thresholds and counters for physical disk power cycles are kept in persistent storage, and are included in state capture data. Informational events are logged when a physical disk is power cycled.

Reset RAID Controller Module

Description

This command resets a RAID controller module.

NOTE: When you reset a RAID controller module, the RAID controller module is not available for I/O operations until the reset is complete. If a host is using virtual disks owned by the RAID controller module being reset, the I/O directed to the RAID controller module is rejected. Before resetting the RAID controller module, either verify that the virtual disks owned by the RAID controller module are not in use or ensure a multipath driver is installed on all hosts using these virtual disks.

Syntax

```
reset controller [(0 | 1)]
```

Parameter

Parameter	Description
`controller`	RAID controller module to reset. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier. If you do not specify an owner, the RAID controller module firmware returns a syntax error.

Additional Information

The RAID controller module that receives the reset RAID controller module command resets the RAID controller module specified. For example, if the reset RAID controller module command is sent to RAID controller module 0 to request a reset of RAID controller module 0, then RAID controller module 0 reboots itself by doing a soft reboot. If the reset RAID controller module command is sent to RAID controller module 0 to request a reset of RAID controller module 1, then RAID controller module 0 holds RAID controller module 1 in reset and then releases RAID controller module 1 from reset, which is a hard reboot. A soft reboot in some products only resets the IOC chip. A hard reboot resets both the IOC chip and the expander chips in the RAID controller module.

Reset Remote Replication Group Statistics

Description

This command resets the synchronization statistics for one or more member virtual disks in a remote replication group to a relative 0.

Syntax

```
reset storageArray arvmStats asyncRemoteReplicationGroup
["asyncRemoteReplicationGroupName"] virtualDisk="virtualDiskName"
sampleType=(all | mostRecent | longestSyncTime | errors)
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group for which you are resetting the synchronization statistics. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets. If the remote replication group name consists only of numbers, such as "1002," you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets.
`virtualDisk`	This parameter is optional. The name of the specific member virtual disk in the remote replication group for which you are resetting the synchronization statistics. If no virtual disk is specified, the statistics for every member virtual disk in the remote replication group are reset. Enclose the virtual disk name in double quotation marks (" ").

Parameter	Description
sampleType	This parameter is optional. The default value for sampleType is all. • all—Data for all three sample types are reset. • mostRecent—Statistics are reset for the most recent 50 resynchronization samples. • longestSyncTime—Statistics are reset for the most recent 20 longest resynchronization samples. • errors—Statistics are reset for the most recent 20 failed resynchronization samples.

Notes

Statistics are reset for replicated virtual disks in the Primary role. The statistics that are reset include the following data:

- Synchronization start time
- Synchronization type (manual or periodic)
- Synchronization duration
- Number of bytes sent
- Maximum and minimum write time (for a single write)
- Maximum and minimum synchronization data rate
- Total write time
- Repository utilization (%)
- Recovery point age

Reset Storage Array Battery Install Date

Description

This command resets the age of the batteries in a storage array to zero days. You can reset the batteries for an entire storage array or just the battery for a specific RAID controller module or in a specific battery pack.

Syntax

```
reset storageArray batteryInstallDate controller=(0 | 1)
```

Parameter

Parameter	Description
controller	Specifies the RAID controller module that contains the battery for which to reset the age. Valid RAID controller module identifier values are 0 and 1. The identifier value is 0 for the RAID controller module on the top and 1 for the RAID controller module on the bottom when viewed from the rear of the

Parameter	Description
	enclosure. Use the `controller` parameter only for RAID controller modules with batteries.

NOTE: If you do not specify a RAID controller module, the age is reset for the storage array battery or both RAID controller module batteries. If you specify a RAID controller module, then the age for only that RAID controller module battery is reset.

Reset Storage Array iSCSI Baseline

Description

This command resets the iSCSI baseline for the storage array to 0.

Syntax

```
reset storageArray iscsiStatsBaseline
```

Parameters

None.

NOTE: This command resets the baseline to 0 for both RAID controller modules in the storage array. The purpose of resetting both of the RAID controller module baselines is to help ensure that the counts are synchronized between the RAID controller modules. If one RAID controller module resets but the second RAID controller module does not reset, the host is informed that the RAID controller modules are out of synchronization. The host is informed by the time stamps that are reported with the statistics.

Example

```
-c "reset storageArray iscsiStatsBaseline;"
```

Reset Storage Array SAS PHY Baseline

Description

This command resets the SAS physical layer (SAS PHY) baseline for all devices except physical disks, and removes the list of errors from the .csv file. The .csv file is generated by running the `save storageArray SASPHYCounts` command.

NOTE: The `reset storageArray SASPHYBaseline` command clears error counts for all devices except the physical disks. After running this command, the .csv file continues to list the **DrivePHY** errors. All other errors are deleted from the .csv file.

Syntax

```
reset storageArray SASPHYBaseline
```

Parameters

None.

Reset Storage Array Virtual Disk Distribution

Description

This command reassigns (moves) all virtual disks to their preferred RAID controller module.

Syntax

```
reset storageArray virtualDiskDistribution
```

Parameters

None.

CAUTION: Ensure that the multipath driver is running before you use this command, or the virtual disk is not accessible.

NOTE: Under certain host operating system environments, you might be required to reconfigure the multipath host physical disk. You might also need to make operating system modifications to recognize the new I/O path to the virtual disks.

Resume Consistency Group Snapshot Virtual Disk

Description

This command restarts a copy-on-write operation for creating a consistency group snapshot virtual disk that you stopped using the `stop cgSnapVirtualDisk` command.

Syntax

```
resume cgSnapVirtualDisk ["snapVirtualDiskName"]
cgSnapImage="snapImageName"
```

Parameters

Parameter	Description
`cgSnapVirtualDisk`	The name of the consistency group snapshot virtual disk for which you want to restart a rollback operation. Enclose the name of the consistency group snapshot virtual disk in double quotation marks (" ") inside square brackets ([ ]).
`cgSnapImage`	The name of the snapshot image in a consistency group that you are restarting. The name of a snapshot image is comprised of two parts: • The name of the consistency group.

Parameter	Description
	• An identifier for the snapshot image in the consistency group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the consistency group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the consistency group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the consistency group. Enclose the snapshot image name in double quotation marks (" ").

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group
- The identifier of the snapshot image

For example, if you want to restart a copy-on-write operation for snapshot image **12345** which is in snapshot consistency group **snapgroup1** in a consistency group snapshot virtual disk that has the name **snapVol1**, you would use this command:

```
resume cgSnapVirtualDisk ["snapVol1"] cgSnapImage=["snapgroup1:12345"]
```

Resume Remote Replication

Description

NOTE: This command valid only with Fibre Channel storage arrays.

This command resumes a suspended Remote Replication operation.

Syntax

```
resume remoteReplication (primary [virtualDiskName] |
primaries [virtualDiskName1 ... virtualDiskNameN])
[writeConsistency=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`primary or primaries`	The name of the primary virtual disk for which you want to resume operation. More than one primary virtual disk name can be entered. Enclose the primary virtual disk name in square brackets ([ ]). If the primary virtual disk name has special

Parameter	Description
	characters, the primary virtual disk name must also be enclosed in double quotation marks (" ").
writeConsistency	The setting to identify the virtual disks in this command that are in a write-consistency group or are separate. For the virtual disks to be in the same write-consistency group, set this parameter to TRUE. For the virtual disks to be separate, set this parameter to FALSE.

NOTE: If you set the writeConsistency parameter to TRUE, the virtual disks must be in a write-consistency group (or groups). This command resumes all write-consistency groups that contain the virtual disks. For example, if virtual disks A, B, and C are in a write-consistency group and they have remote counterparts A', B', and C', the resume remoteReplication virtual disk ["A"] writeConsistency=TRUE command resumes A-A', B-B', and C-C'.

Resume Remote Replication Group

Description

This command resumes data transfer between all replicated pairs in a remote replication group. Data written to the primary virtual disks while the remote replication group was suspended is written to the secondary virtual disks immediately. Periodic synchronization resumes if an automatic synchronization interval has been set.

Syntax

```
resume asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
[deleteRecoveryPointIfNecessary]
```

Parameter

Parameter	Description
asyncRemoteReplicationGroup	The name of the remote replication group for which you want to start synchronization. Enclose the remote replication group name in double quotation marks (" ") inside square brackets ([ ]").
deleteRecoveryPointIfNecessary	The parameter to delete the recovery point if the recoverable synchronization data has exceeded time threshold for recovery. Recovery point age is measured from the time the data was frozen on the primary storage array.

Resume Snapshot Image Rollback

Description

This command resumes a rollback operation that has entered a paused state. A rollback operation can enter a paused state due to processing errors, which causes a Needs Attention condition for the storage array.

If the rollback operation cannot be resumed, the selected snapshot virtual disk reverts to a paused state, and the Needs Attention condition is displayed.

NOTE: You cannot use this command for snapshot images involved in online virtual disk copy.

Syntax

```
resume snapImage [snapImageName] rollback
```

Parameter

Parameter	Description
`snapImage`	The name of the snapshot image for which you want to restart a rollback operation. The name of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]).

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group
- The identifier of the snapshot image

For example, if you want to restart a rollback operation for snapshot image **12345** in a snapshot group that has the name **snapgroup1**, you would use this command:

```
resume snapImage ["snapgroup1:12345"] rollback;
```

Resume Snapshot (Legacy) Rollback

Description

NOTE: With firmware version 7.83 the order of the terms in the syntax is changed to be consistent with other commands. Replace the deprecated command syntax with the new command syntax.

This command resumes a rollback operation that has entered a paused state. A rollback operation can enter a paused state due to processing errors, which triggers a **Needs Attention** condition for the storage array.

If the rollback operation cannot be resumed, the selected snapshot (legacy) virtual disk reverts to a paused state, and the **Needs Attention** condition is displayed.

Syntax

Deprecated

```
resume rollback virtualDisk [snapshotVirtualDiskName]
```

New

```
resume virtualDisk [snapshotVirtualDiskName] rollback
```

Parameter

Parameter	Description
`virtualDisk`	Specifies the name of the snapshot (legacy) virtual disk for which you want to resume a rollback operation. Enclose the snapshot (legacy) virtual disk name in square brackets ([ ]). If the snapshot (legacy) virtual disk name has special characters, you must also enclose the snapshot (legacy) virtual disk name in double quotation marks (" ").

Resume Snapshot Virtual Disk

Description

This command resumes a snapshot virtual disk operation that was stopped.

Syntax

```
resume snapVirtualDisk [snapVirtualDiskName] snapImage="snapImageName"
```

Parameters

Parameter	Description
`snapVirtualDisk`	The name of the specific snapshot virtual disk for which you want to resume a rollback operation. Enclose the snapshot virtual disk identifier in square brackets ([ ]). If the snapshot virtual disk identifier has special characters, you must also enclose the snapshot virtual disk identifier in double quotation marks (" ").
`snapImage`	The name of the snapshot image for which you want to resume snapshot virtual disk operations. The name of a snapshot image is comprised of two parts: • The name of the snapshot group • An identifier for the snapshot image in the snapshot group The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]).

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group
- The identifier of the snapshot image

For example, if you want to resume snapshot virtual disk operations for snapshot image **12345** in a snapshot group that has the name **snapGroup1**, use the following command:

```
resume snapVirtualDisk [snapVol1] snapImage=["snapGroup1:12345"];
```

To resume snapshot virtual disk operations for the most recent snapshot image in a snapshot group that has the name **snapGroup1**, use the following command:

```
resume snapVirtualDisk [snapVol1] snapImage=["snapGroup1:newest"];
```

Resume SSD Cache

Description

This command restarts the caching for all virtual disks using the SSD cache that was temporarily stopped with the `suspend ssdCache` command.

Syntax

`resume ssdCache [ssdCacheName]`

Parameter

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache that you want to resume. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Revive Disk Group

Description

This command forces the specified disk group and associated failed physical disks to the Optimal state. All physical disks assigned to the disk group must be installed before you attempt to run this command.

CAUTION: Correct use of this command depends on the data configuration on all of the physical disks in the disk group. Never attempt to revive a physical disk unless supervised by a Customer or Technical Support representative.

Syntax

`revive diskGroup [diskGroupNumber]`

Parameter

Parameter	Description
`diskGroup`	Number of the disk group to be set to the optimal state. You must put brackets ([ ]) around the disk group number.

Revive Disk Pool

Description

This command forces a disk pool and its associated physical disks to the Optimal state.

CAUTION: Correct use of this command depends on the data configuration on all of the physical disks in the disk pool. Do not try to revive a disk pool unless you are supervised by your Technical Support representative.

Syntax

```
revive diskPool [diskPoolName]
```

Parameter

Parameter	Description
`diskPool`	The alphanumeric identifier (including - and _) of the disk pool that you want to revive. Enclose the *diskPoolName* in square brackets ([ ]).

Revive Physical Disk

Description

This command forces the specified physical disk to the optimal state.

CAUTION: Correct use of this command depends on the data configuration on all physical disks in the disk group. Never attempt to revive a physical disk unless supervised by a Technical Support representative.

Syntax

```
revive physicalDisk [enclosureID,slotID]
```

Parameter

Parameter	Description
`physicalDisk`	Identifies the enclosure and slot where, the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put brackets ([ ]) around the enclosure ID value and the slot ID value.

Revive Snapshot Group

Description

This command forces the specified snapshot group to the Optimal state. If the snapshot group is not in a Failed state, the firmware displays an error message and does not run this command.

Syntax

```
revive snapGroup [snapGroupName]
```

Parameter

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group that you want to set to the Optimal state. Enclose the snapshot group identifier in square brackets ([ ]).

Revive Snapshot Virtual Disk

Description

This command forces a specified snapshot virtual disk to the **Optimal** state. The snapshot virtual disk can be:

- A standalone snapshot virtual disk
- A snapshot virtual disk that is a member of a consistency group

If the snapshot virtual disk is not in a **Failed** state, the firmware displays an error message and does not run this command.

NOTE: You cannot use this command for a snapshot virtual disk that is used in online virtual disk copy.

Syntax

```
revive snapVirtualDisk [snapVirtualDiskName]
```

Parameter

Parameter	Description
`snapVirtualDisk`	The alphanumeric identifier (including - and _) of the snapshot virtual disk that you want to set to the Optimal state. Enclose the snapshot virtual disk identifier in square brackets ([ ]).

Save Enclosure Log Data

Description

This command saves the enclosure log sense data to a file. Log sense data is maintained by the environmental cards for each enclosure. Not all of the environmental cards contain log sense data.

Syntax

```
save allEnclosures logFile="filename"
```

Parameter

Parameter	Description
`logFile`	Name of the file to which to save the log sense data. You must put quotation marks (" ") around the file name.

Save Input Output RAID Controller Module (IOC) Dump

This command saves the IOC dumps from the RAID controller modules in a storage array to a file on a host.

Syntax

save IOCLog [file="filename"]

Parameter

Parameter	Description
file	The file path and the file name to which you save the IOC dump. Enclose the file name in double quotation marks (" "). For example: file="C:\Array Backups\IOCcoredump1.7z" This command saves the data to a compressed file and adds the file extension .7z to the saved file. The default name of the file that contains the IOC-related data for RAID controller modules in the storage array uses the WWN of the storage array.

NOTE: This command retrieves from both RAID controller modules the IOC dump log data and the IOC dump metadata. Using a 7zip file format, the retrieved data is archived and compressed into a single file with a file name of your choice. The 7zip archive file contains the following:

filename+"IOCLog"+[A|B].gz – the IOC logs retrieved from RAID controller module 0 or RAID controller module 1 if available

filename+"IOCLogInfo"+[A|B].txt – the IOC logs metadata info retrieved from RAID controller module 0 or RAID controller module 1. If the IOC log data cannot be retrieved from a RAID controller module, the IOC log data cannot be retrieved from a RAID controller module, the metadata .txt file will contain the error condition and reason.

Following are the error conditions:

The RAID controller module platform and HIC do not support IOC dump.

The RAID controller modules have not collected IOC dump data.

The compressed logs are not in a human-readable format. You must return the logs to Technical Support to be evaluated.

Save Physical Disk Log

Description

This command saves the log sense data to a file. Log sense data is maintained by the storage array for each physical disk.

Syntax

```
save allPhysicalDisks logFile="filename"
```

Parameter

Parameter	Description
`logFile`	Name of the file to which to write the log sense data. You must put quotation marks (" ") around the file name.

Save RAID Controller Module NVSRAM

Description

This command saves a copy of the RAID controller module NVSRAM values to a file. This command saves all regions.

Syntax

```
save controller [(0 | 1)] NVSRAM file="filename"
```

Parameters

Parameter	Description
`controller`	RAID controller module with the NVSRAM values to save. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier.
`file`	Name of the file in which to save the values. You must put quotation marks (" ") around the file name.

Save Remote Replication Group Statistics

Description

This command saves to a file the synchronization statistics for one or more member virtual disks in a remote replication group. The statistics collected are available only for remote replication group member virtual disks in a primary role on the local storage array.

A set of data is collected during the synchronization process that can be used to evaluate how well the replication configuration is functioning. The data is collected as a set of samples. A sample is created at the beginning of a synchronization process and updated regularly while the synchronization process proceeds.

A sample collects data until the synchronization process completes or until a disruption to the synchronization process occurs such as a virtual disk ownership transfer or a read-write error. When a synchronization process disruption is resolved (for example, the virtual disk is transferred to the alternate RAID controller module), a new sample is created and updated as the synchronization process continues.

Syntax

```
save storageArray arvmStats asyncRemoteReplicationGroup
["asyncRemoteReplicationGroupName"]
arvmStats file="filename"
[virtualDisk="virtualDiskName"
sampleType=(all | mostRecent | longestSyncTime | errors)
recordLimit=(1-90)]
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group for which you are retrieving synchronization statistics. Enclose the remote replication group name in square brackets ([ ]). If the remote replication

Parameter	Description
	group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets. If the remote replication group name consists only of numbers, such as "1002," you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets.
`file`	The file path and the file name to which you want to save the synchronization statistics. Enclose the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI\logs \iscsistat.csv"** This command does not automatically append a file extension to the saved file. You can use any file name, but you must use the **.csv** extension.
`virtualDisk`	This parameter is optional. The name of the specific member virtual disk in the remote replication group for which you are retrieving synchronization statistics. If no virtual disk is specified, the statistics for every member virtual disk in the remote replication group are saved. If statistics are collected for more than one virtual disk in a remote replication group, all the data will be written to the same file. Enclose the virtual disk name in double quotation marks (" ").
`sampleType`	This parameter is optional. The default value for `sampleType` is `all`. • `all`—Data for all three sample types are collected and written to the same file. • `mostRecent`—Statistics are recorded for the most recent 50 resynchronization samples. • `longestSyncTime`—Statistics are collected for the most recent 20 longest resynchronization samples. • `errors`—Statistics are recorded for the most recent 20 failed resynchronization samples. These samples include a failure code.
`recordLimit`	This parameter is optional. The default value for record limit is `no limit`. The `recordLimit` must be greater than 0 and less than or equal to 90.

Notes

Statistics are captured for replicated virtual disks in the Primary role. The statistics collected include the following data:

- Synchronization start time
- Synchronization type (manual or periodic)
- Synchronization duration
- Number of bytes sent
- Maximum write time (for a single write)
- Minimum write time (for a single write)
- Minimum synchronization data rate
- Maximum synchronization data rate
- Total write time
- Repository utilization (%)
- Recovery point age

During initial synchronization, statistics samples are captured approximately every 15 minutes.

The synchronization statistics are included in the support bundle.

Save Storage Array Configuration

Description

This command creates a script file that you can use to create the current storage array virtual disk configuration.

Syntax

```
save storageArray configuration file="filename"
[(allConfig | globalSettings=(TRUE | FALSE) virtualDiskConfigAndSettings=(TRUE
| FALSE)
hostTopology=(TRUE | FALSE)
lunMappings=(TRUE | FALSE))]
```

Parameters

Parameter	Description
`file`	Name of the file that contains the configuration values. You must put quotation marks (" ") around the file name.
`allConfig`	Saves all of the configuration values to the file. (If you choose this parameter, all of the configuration parameters are set to `TRUE`.)
`globalSettings`	Saves the global settings to the file. To save the global settings, set this parameter to `TRUE`. To

Parameter	Description
	prevent saving the global settings, set this parameter to `FALSE`. The default value is `TRUE`.
`virtualDiskConfigAndSettings`	Saves the virtual disk configuration settings and all of the global settings to the file. To save the virtual disk configuration and global settings, set this parameter to `TRUE`. To prevent saving the virtual disk configuration and global settings, set this parameter to `FALSE`. The default value is `TRUE`.
`hostTopology`	Saves the host topology to the file. To save the host topology, set this parameter to `TRUE`. To prevent saving the host topology, set this parameter to `FALSE`. The default value is `FALSE`.
`lunMappings`	Saves the LUN mapping to the file. To save the LUN mapping, set this parameter to `TRUE`. To prevent saving the LUN mapping, set this parameter to `FALSE`. The default value is `FALSE`.

NOTE: When you use this command, you can specify any combination of the parameters for global setting, virtual disk configuration setting, host topology, or LUN mapping. To enter all settings, use the `allConfig` parameter. The parameters are optional; you do not have to enter any parameters.

Save Storage Array Core Dump

Description

This command saves a core dump to a specified file on a host.

Syntax

```
save storageArray coreDump file="filename"
```

Parameter

Parameter	Description
`file`	The file path and the file name to which you save the core dump. Enclose the file name in double quotation marks (" "). For example: `file="C:\Array Backups\DBMbackup_03302010.core"`

Save Storage Array DBM Database

Description

This command saves the database management (DBM) of the storage array.

Syntax

```
save storageArray dbmDatabase
[sourceLocation=(disk | onboard) |
controller [(0|1)] |
contentType=(raidConfigration | all)]
file="fileName"
```

Parameters

Parameter	Description
`sourceLocation`	This parameter specifies the location from which to obtain backup database information. • `disk` indicates that data comes directly from the database on the physical disk • `cache` indicates that data comes from the RPA memory location The default location is `disk`.
`controller`	This parameter specifies the RAID controller module from which data will be exclusively retrieved, if `sourceLocation` is set to `onboard`. If the `controller` parameter is not specified, data might be retrieved from either RAID controller modules. Valid RAID controller module identifiers are `0` or `1`, where `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1.
`contentType`	This parameter specifies the content type of the data that will be retrieved. • If the parameter is set to `raidConfiguration`, the RAID configuration data will be retrieved. The RAID configuration data includes disk groups and associated virtual disks. • If the parameter is set to `all`, all data that resides in the database will be retrieved. This includes disk group and disk pool configuration. The default is `raidConfiguration`.
`file`	The file path and the file name to which you want to save the database. Enclose the file name in double quotation marks (" "). For example: `file="C:\Program Files\CLI\logs\events.txt"`. This parameter must appear last, after any of the optional parameters.

Parameter	Description
	This command does not automatically append a file extension to the saved file. You must specify a file extension when entering the file name.

Notes

The data that you save to the host using this command can, if needed, be restored to the RAID controller module. A validator, however, is required to restore data from the file on the host.

Save Storage Array DBM Validator Information File

Description

This command saves the database management (DBM) validation information for a storage array in an XML file. Support personnel can use this XML file to generate a validator string (a security code). The validator string must be included in the `load storageArray dbmDatabase` command when restoring a storage array back to a pre-existing configuration.

Syntax

`save storageArray dbmValidatorInfo file="`*`filename`*`" dbmDatabase="`*`filename`*`"`

Parameters

Parameter	Description
`file`	The file path and the file name of the DBM validator. Enclose the file name in double quotation marks (" "). For example: **file="C:\Array Backups \DBMvalidator.xml"**. This command does not automatically append a file extension to the saved file. You must specify the **.xml** file extension when you enter the file name.
`dbmDatabase`	The file path and the file name of the DBM database from which you want to generate the XML information file. Enclose the file name in double quotation marks (" "). For example: **dbmDatabase="C:\Array Backups \DBMbackup_03302010.dbm"**. This command does not automatically append a file extension to the saved file. You must specify the **.dbm** file extension when you enter the file name.

Notes

If you are generating the XML information file in order to obtain a validator string, you must run this command while you are connected to the RAID controller module where you intend to restore the database. The following example show the format of the XML file:

```
<?xml version="1.0" encoding="utf-8"?>
<DbmImageMetadata>
<Controllers>
<A>1IT0nnnnnnnnnABCDA>1IT0nnnnnnnnnABCD<A/>
<B>1T0nnnnnnnnnABCDB</B>
</Controllers>
<Header>
<ImageVersion>1</ImageVersion>
<TimeStamp>1269388283</TimeStamp>
</Header>
<Trailer>
<CRC>nnnnnnnnnn</CRC>
</Trailer>
</DbmImageMetadata>
```

Save Storage Array Events

Description

This command saves events from the Major Event Log (MEL) to a file.

Any or all of the following events can be saved:

- Critical events—Errors occurring on the storage array that needs to be addressed immediately. Loss of data access may occur if the error is not immediately corrected.
- Warning events—Errors occurring on the storage array resulting in degraded performance or reduced ability to recover from additional errors. Access to data has not been lost, but the must be corrected to prevent possible loss of data access in the event of an additional error.
- Informational events—Events occurring on the storage array that do not impact normal operations. This event is reporting a change in configuration or other information useful in evaluating the performance of the storage array.
- Debug events—Events occurring on the storage array that provides information useful in determining steps or states that led to the error. This information may be useful to your Technical Support representative in helping determine error causes.

NOTE: Some storage arrays may be unable to support all the above mentioned types of events.

Syntax

```
save storageArray (allEvents | criticalEvents | warningEvents | infoEvents |
debugEvents) file="filename" [count=numberOfEventsforceSave=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`allEvents`	Saves all events to a file.
`criticalEvents`	Saves only critical events to a file.

Parameter	Description
warningEvents	Saves only warning events to a file.
infoEvents	Saves only informational events to a file.
debugEvents	Saves only debug events to a file.
file	Specifies name of the file to which to save the events. You must put quotation marks (" ") around the file name.
count	Specifies the number of events or critical events to save to a file. If you do not enter a value for the count, all events or critical events are saved to the file. If you enter a value for the count, only that number of events or critical events (starting with the last event entered) are saved to the file. Use integer values.
forceSave	When set to TRUE, force saves all critical events to a file. Default value is FALSE.

Save Storage Array iSCSI Statistics

Description

This command saves the iSCSI performance of the storage array to a file. The following statistics are saved to the file:

- Statistics related to the physical Ethernet port
- Statistics related to the TCP protocol
- Statistics related to the IP protocol

Syntax

```
save storageArray iscsiStatistics [(raw | baseline)] file="filename"
```

Parameters

Parameter	Description
raw	This parameter defines that the statistics collected are all statistics from the RAID controller module start-of-day. Enclose the parameter in square brackets ([ ]).
baseline	This parameter defines that the statistics collected are all statistics from the time the RAID controller modules were reset to zero using the reset storageArray iscsiStatsBaseline command. Enclose the parameter in square brackets ([ ]).

Parameter	Description
`file`	The name of the file to which you want to save the performance statistics. Enclose the file name in double quotation marks (" "). NOTE: This command does not automatically append a file extension to the new file. You must specify the file extension when you enter the file name.

NOTE: If you have not reset the iSCSI baseline statistics since the RAID controller module start-of-day, the time at the start-of-day is the default baseline time.

Save Storage Array Performance Statistics

Description

This command saves the performance statistics to a file. Before you use this command, issue the `set session performanceMonitorInterval` and `set session performanceMonitorIterations` commands to specify how often statistics are collected.

Syntax

```
save storageArray performanceStats file="filename"
```

Parameters

Parameter	Description
`file`	Specifies name of the file to which to save the performance statistics. You must put quotation marks (" ") around the file name.

Save Storage Array SAS PHY Counts

Description

This command saves the storage array SAS physical layer (SAS PHY) counters to a file. To reset the SAS PHY counters, run the `Reset Storage Array SAS PHY Baseline` command. See Reset Storage Array SAS PHY Baseline.

Syntax

```
save storageArray SASPHYCounts file="filename"
```

Parameters

Parameter	Description
`file`	Specifies the file path and the file name to which you save the SAS PHY counters. Enclose the file path and the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI\logs \sasphy.csv"** This command does not automatically append a file extension to the saved file. Any file name is acceptable, but the .**csv** extension must be used.

Save Storage Array State Capture

Description

This command saves the state capture to a file.

Syntax

```
save storageArray stateCapture file="filename"
```

Parameters

Parameter	Description
`file`	Name of the file to which to save the state capture. You must put quotation marks (" ") around the file name.

Save Storage Array Support Data

Description

This command saves the storage array support-related information to a file.

Support-related information includes:

- The storage array profile
- The Major Event Log information
- The NVSRAM data
- Current problems and associated recovery information
- The performance statistics for the entire storage array
- The persistent registration information and the persistent reservation information
- Detailed information about the current status of the storage array
- The diagnostic data for the physical disk
- A recovery profile for the storage array

- The unreadable sectors that are detected on the storage array
- The state capture data
- An inventory of the versions of the firmware running on the RAID controller modules, the physical disks, and the RAID controller modules

Syntax

```
save storageArray supportData file="filename"
[force=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`file`	The path and the file name where support-related data for the storage array is saved. Enclose the file path and the file name in double quotation marks (" "). For example, `file="C:\Program Files\CLI \logs\supdat.zip".` This command does not automatically append a file extension to the saved file. A file extension must be specified when entering the file name.
`force`	Forces collection of support data when there is a failure in securing a lock on RAID controller module operations. To force the collection of support data, set this parameter to `TRUE`. The default value is `FALSE`.

Schedule Support Bundle Collection Configuration

Description

This command specifies a schedule for collecting support bundle data.

NOTE: This command must be run from the command line. It cannot be run in the MD Storage Manager script editor.

Syntax

```
SMcli -supportBundle schedule (enable|disable) (all|storageArrayName)
[data=pathName]
[startTime=HH:MM]
[startDate=MM:DD:YY]
[endDate=MM:DD:YY]
(daysOfWeek={} | months={}
(onDays={} |
weekNumber=(First|Second|Third|Fourth|Last) dayOfWeek=(Sunday|Monday|Tuesday|
Wednesday|)
```

Parameters

Parameter	Description
`enable`	Enable automatic collection of support bundles when a critical MEL event is detected.
`disable`	Disable automatic collection of support bundles where the feature was previously enabled.
`all`	Apply the command to all storage arrays detected from the host.
`storageArrayName`	Apply the command to the named storage array.
`data`	Store the data generated by the command at the location specified by the *pathName*.

Scheduling Snapshots

NOTE: This command is applicable to both the Snapshot and the Snapshot (Legacy) premium feature.

Description

You can use the `enableSchedule` and the `schedule` parameters to schedule automatic snapshots. Automatic snapshots can be scheduled daily, weekly, or monthly (by day or by date). The `enableSchedule` parameter turns on or off the ability to schedule snapshots. When you enable scheduling, you use the `schedule` parameter to define when you want the snapshots to occur.

Parameters

Parameter	Description
`immediate`	Begins a snapshot operation immediately; a snapshot virtual disk is created and a copy-on-write operation begins.
`startDate`	Defines a specific start date snapshot virtual disk creation. The format for this option is `MM:DD:YY`. If no start date is provided, the current date is used. For example: `startDate=06:27:11`.
`scheduleDay`	Defines a day of the week you want to create a snapshot virtual disk and perform a copy-on-write operation. The values that you can enter are: `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, `sunday`, and all. For example: `scheduleDay=wednesday`.
`startTime`	Defines time of a day you want to create a snapshot virtual disk and start performing a copy-on-write operation. The format for entering the

Parameter	Description
	time is `HH:MM`, where `HH` is the hour and `MM` is the minute past the hour. Use a 24-hour clock. For example: `startTime=14:27`, which corresponds to 2:27 PM.
`scheduleInterval`	Amount of time, in minutes that defines a minimum between two copy-on-write operations. It is possible for you to create a schedule in which you have overlapping copy-on-write operations because of the duration of a copy operation. You can ensure that copy-on-write operations maintain a correct interval by using this option. The maximum value for the `scheduleInterval` option is 1440 minutes. For example: `scheduleInterval=180`.
`endDate`	Specific date to stop creating a snapshot virtual disk and end the copy-on-write operations. The format is `MM:DD:YY`, where `MM` is the month, `DD` the day and `YY` the year. For example: `endDate=11:26:11`.
`noEndDate`	Use this option to prevent a scheduled copy-on-write operation from ending. If you later decide to end the copy-on-write operations, you must re-enter the create `snapshotVirtualDisk` command and specify an end date.
`timesPerDay`	Number of times you want the snapshot schedule to run in a day. For example, `timesPerDay=4`.
`timeZone`	Specifies the time zone to use for the scheduling operations, using one of the following values: • `GMT±HH:MM`—The offset from Greenwich Mean Time (GMT), in hours and minutes. For example, GMT-06:00 must be used to specify Central Standard Time (CST). • `Text string`—A standard time zone text string, in quotation marks. For example, "USA/Chicago" or "Germany/Berlin". Time zone text strings are case sensitive. If an incorrect text string is used, or no text string is specified, the default time zone is GMT. For example, CST can be specified in two different ways: `timeZone=GMT-06:00` or `timeZone="USA/Chicago`.

NOTE: To remove a schedule, use the `delete virtualDisk` command with the `schedule` parameter. The `delete` command with the `schedule` parameter deletes only the schedule, not the snapshot virtual disk. Ensure that you do not inadvertently delete the virtual disk itself.

NOTE: If you also use the `scheduleInterval` option, the lowest value of the `timesPerDay` option and the `scheduleInterval` option is used. The value is determined by calculating an integer value for the `scheduleInterval` option. Divide 1440 by the `scheduleInterval` option value that you set. For example, 1440/180 = 8. This value is then compared to the `timesPerDay` integer value and the calculated `scheduleInterval` integer value; the smaller value is then used.

Set Consistency Group Attributes

Description

This command defines the properties for a snapshot consistency group.

Syntax

```
set consistencyGroup ["consistencyGroupName"]
[userLabel="consistencyGroupName"] [repositoryFullPolicy=(failBaseWrites |
purgeSnapImages)] [repositoryFullLimit=percentValue]
[autoDeleteLimit=numberOfSnapImages] [rollbackPriority=(lowest | low | medium |
high | highest)]
```

Parameters

Parameter	Description
`consistencyGroupName`	The name of the consistency group for which you are setting properties. Enclose the consistency group name in double quotation marks (" ") inside square brackets ([ ]).
`userLabel`	The new name that you want to give to the snapshot consistency group. Enclose the new consistency group name in double quotation marks (" ").
`repositoryFullPolicy`	How you want snapshot processing to continue if the snapshot repository virtual disks are full. You can choose to fail writes to the base virtual disk (`failBaseWrites`) or delete (purge) the snapshot images (`purgeSnapImages`). The default action is `purgeSnapImages`.
`repositoryFullLimit`	The percentage of repository capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.
`autoDeleteLimit`	The maximum number of snapshot images that you want to automatically delete if you have selected to purge the snapshot images for a repository full policy. The default value is `32`.

Parameter	Description
rollBackPriority	The priority for rollback operations for a consistency group while the storage array is operational. Valid values are highest, high, medium, low, or lowest.

Set Consistency Group Snapshot Virtual Disk

Description

This command creates a unique name for a snapshot virtual disk of a consistency group.

Syntax

```
set cgSnapVirtualDisk [consistencyGroupName]
userLabel="snapVirtual DiskName"
```

Parameters

Parameter	Description
cgSnapVirtualDisk	The name of the consistency group virtual disk that you want to rename. Enclose the name of the consistency group snapshot virtual disk in square brackets ([ ]).
userLabel	The new name that you want to give to the snapshot virtual disk in the consistency group. Enclose the new snapshot virtual disk name in double quotation marks (" ").

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

Set Disk Group

Description

This command defines the properties for a disk group.

Syntax

```
set diskGroup [diskGroupName]
addPhysicalDisks=(enclosureID1,slotID1 enclosureID2,slotID2...
enclosureIDn,slotIDn)
raidLevel=(0 | 1 | 5 | 6)
owner=(0 | 1)
```

Parameters

Parameter	Description
`diskGroup`	Sequence number of the disk group for which to set properties. You must put brackets ([ ]) around the disk group number.
`addPhysicalDisks`	Identifies the physical disk by enclosure and slot location to include in the disk group. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put parentheses around the enclosure ID values and the slot ID values.
`raidLevel`	The RAID level for the disk group. Valid values are `0`, `1`, `5`, or `6`.
`owner`	The RAID controller module that owns the disk group. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. Use this parameter only if you want to change the disk group owner.

NOTE: When using this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters. Using too many parameters can cause host I/O errors or result in internal RAID controller module reboots because the time period ends before the disk group definition is set.

NOTE: For configurations with more than 32 virtual disks, it is recommended you quiesce host I/O operations when executing this command.

Troubleshooting

Attempting to expand large disk groups by adding physical disks, also called Dynamic Capacity Expansion (DCE), may fail with the one of the following messages:

- `Return code: Error 26—The modification operation cannot complete because of the number of physical disks in the disk group and the segment size of the associated virtual disks. Reduce the segment size of all virtual disks in the disk group to 128 KB or below using the Change Segment Size option. Then, retry the operation.`
- `Return code: Error 462—A SYMbol procedure could not be carried out because the firmware could not allocate sufficient cache memory. Operation when error occurred: PROC_startVolum.`

  In addition, a Major Event Log (MEL) event indicating insufficient cache available to complete the DCE operation may occur.

DCE requires enough memory to buffer the data read from the original virtual disk and the data to be written to the expanded virtual disk. Some combination of number of physical disks in the expansion

operation, stripe size, and whether replication is enabled may result in less memory being available to complete the DCE operation. In such a scenario, try each of the options below and retry the operation:

- Create the required disk group size using other unassigned physical disks.
- Delete the current disk group and then recreate the disk group with the desired number of physical disks.
- Reduce the segment size being used and then retry the operation.

Set Disk Group Forced State

Description

This command moves a disk group into a Forced state. Use this command if the `start diskGroup import` command does not move the disk group to an Imported state or if the import operation does not work because of hardware errors. In a Forced state, the disk group can be imported, and you can then identify the hardware errors.

Syntax

```
set diskGroup [diskGroupName] forcedState
```

Parameters

Parameter	Description
`diskGroup`	Sequence number of the disk group for which to set properties. You must put brackets ([ ]) around the disk group number.

NOTE: You can move the physical disks that comprise a disk group from one storage array to another storage array. The CLI provides three commands that let you move the physical disks. The commands are `start diskGroup export`, `start diskGroup import`, and `set diskGroup forcedState`. In the Forced state, you can perform an import operation on the disk group.

Set Disk Pool

Description

This command sets the attributes associated with a disk pool based on the specified parameters.

Syntax

```
set (diskPool [diskPoolName] | diskPools [diskPoolName1... diskPoolNamen] |
allDiskPools) [reservedPhysicalDiskCount=reservedPhysicalDiskCountValue]
[warningThreshold=(warningThresholdValue | default)]
[criticalThreshold=(criticalThresholdValue | default)]
[criticalPriority=(highest | high | medium |low |lowest)]
[degradedPriority=(highest |high |medium |low |lowest)]
[backgroundPriority=(highest | high | medium |low |lowest)]
[userLabel=diskPoolName]
```

Parameters

Parameter	Description
`diskPools`	The list of *diskPoolNames* identifies multiple physical disk pool which have their parameters reset by the command.
`allDiskPools`	This parameter specifies that all physical disk pools have their parameters reset by the command.
`reservePhysicalDiskCount`	This value is specified in terms of the number of physical disks to reserve in the disk pool for physical disk failures or physical disk removals.
`warningThreshold`	The percentage of disk pool capacity at which you receive a warning alert that the disk pool is nearing full capacity. Use integer values. For example, a value of `70` means 70 percent. For best operation, the value for this parameter must be less than the value for the `criticalThreshold` parameter. Setting this parameter to `0` disables warning alerts.
`criticalPriority`	This parameter governs the priority for critical reconstruction activities on the disk pool.
`backgroundPriority`	This parameter governs the priority for background task activities on the disk pool.

Set Disk Pool Complete

Description

This command sets a disk pool that is in an Incomplete state to the Complete state.

Syntax

`set diskPool [`*`diskPoolName`*`] complete`

Parameter

Parameter	Description
`diskPool`	The alphanumberic identifier of the disk pool (including – and _) that you want to set to a Complete state. Enclose the *diskPoolName* in square brackets ([ ]).

Set Enclosure Attribute

Description

This command sets user-defined attributes for an enclosure.

Syntax

```
set enclosure [enclosureID] (chassisName | assetTag)= "userID"
```

Parameters

Parameter	Description
enclosureID	Identifies a specific enclosure for which to set the attribute. Enclosure ID values are 0 to 99. You must put brackets ([ ]) around the enclosure ID value.
chassisName	Chassis name or number to give the new enclosure. Chassis names can be any combination of alphanumeric characters with a maximum length of 32 characters. Alphabetical characters can be uppercase or lowercase. You can also use the underscore character (_) and the hyphen (-) character. You cannot use spaces in a chassis name. You must put quotation marks (" ") around the chassis name.
assetTag	Asset tag name or number to give the new enclosure. Asset tags can be any combination of alphanumeric characters with a maximum length of ten characters. Alphabetical characters can be uppercase or lowercase. You can also use the underscore character (_) and the hyphen (-) character. You cannot use spaces in an asset tag name. You must put quotation marks (" ") around the asset tag name.

Set Enclosure Identification

Description

This command sets the ID of an enclosure in a storage array.

Syntax

```
set enclosure ["Service Tag"] id=enclosureID
```

Parameters

Parameter	Description
`enclosure`	Service tag of the RAID enclosure or the expansion enclosure for which you are setting the enclosure ID. You must put quotation marks (" ") around the Service Tag.
`id`	Specifies the value for the RAID enclosure or expansion enclosure ID. Valid values are `0` through `99`. You do not need to put parentheses around the enclosure ID value.

Set Event Alert Filtering

Description

The set event alert commands manage alert event notification by either disabling a specific critical event alert from being sent, or by enabling notification about a specific critical event alert to be sent. To prevent notification about a specific critical event alert from being sent, you block it. To enable notification about a specific critical event alert to be sent, you unblock it.

The storage management software supports different priorities of events (such as Informational, Warning, Critical). When you define your storage array, you can configure alerts and define how the event alerts are managed. If you have configured the storage array to send alerts, then a notification is sent to a designated recipient when a Critical event occurs. That notification can be one or all of the following types:

- email
- syslog
- trap notifications

The set event alert CLI commands work on a single storage array. When you run the commands on a storage array, only that storage array is affected by the commands. Other storage arrays that have not had the CLI command run against them have the default behavior.

Blocking an event alert does not prevent the event from being posted to the system event log. All events will continue to be posted to the event log.

Syntax

Syntax to Block an Event Alert

```
set blockEventAlert eventType
```

Syntax to Unblock an Event Alert

```
set unBlockEventAlert eventType
```

Parameters

Parameter	Description
`eventType`	This parameter is the integer value for the event. Enter the event value in a hexadecimal format, for example, 0x280D. Always begin the hexadecimal value with **0x** to indicate that it is in hexadecimal format. If you do not use 0x, the value will be interpreted as a decimal and converted to a hexadecimal value, before applying the block or unblock command. This can cause an incorrect event to be blocked or unblocked. An error is displayed if you enter an invalid event.

Set Foreign Physical Disk to Native

A physical disk is considered to be native when it is a part of a storage array disk group. A physical disk is considered to be foreign when it does not belong to a storage array disk group or when it fails to be imported with the physical disks of a disk group that are transferred to a new storage array. The latter failure creates an incomplete disk group on the new storage array.

Description

Run this command to add the missing (foreign) physical disks back into their original disk group and to make them part of the new storage array disk group configuration of the new storage array.

Use this operation for emergency recovery only: when one or more physical disks must be changed from a foreign physical disk status and returned to a native status within their original disk group.

CAUTION: Using this command for purposes other than what is stated above might result in data corruption or data loss without notification.

Syntax

```
set (physicalDisk [enclosureID,slotID] |
physicalDisks [enclosureID1,slotID1 enclosureID2, slotID2...
enclosureIDn,slotIDn]|
allPhysicalDisks) nativeState
```

Parameters

Parameter	Description
`physicalDisk or physicalDisks`	The enclosure and the slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. Enclose the enclosure ID values and the slot ID values in square brackets ([ ]).
`allPhysicalDisks`	Selects all the physical disks.

Set Host

Description

This command assigns a host to a host group or moves a host to a different host group. You can also create a new host group and assign the host to the new host group with this command. The actions performed by this command depend on whether the host has individual mappings or does not have individual mappings.

Syntax

```
set host [hostName]
hostGroup=("hostGroupName" | none | defaultGroup)
userLabel="newHostName"
hostType=(hostTypeIndexLabel | hostTypeIndexNumber)
```

Parameters

Parameter	Description
`host`	The name of the host that you want to assign to a host group. Enclose the host name in square brackets ([ ]). If the host name has special characters, you must also enclose the host name in double quotation marks (" ").
`hostGroup`	The host group to which you want to assign the host. (The following table defines how the command runs if the host does or does not have individual virtual disk-to-LUN mappings.) Enclose the host group name in double quotation marks (" "). The **defaultGroup** is the host group that contains the host to which the virtual disk is mapped.
`userLabel`	The new host name. Enclose the host name in double quotation marks (" ").
`hostType`	The index label or number of the host type for the HBA host port. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers. If the host type has special characters, enclose it in double quotation marks (" ").

Host Group Parameter	Host Has Individual Virtual Disk-to-LUN Mappings	Host Does Not Have Individual Virtual Disk-to-LUN Mappings
hostGroupName	The host is removed from the present host group and is placed	The host is removed from the present host group and is placed

Host Group Parameter	Host Has Individual Virtual Disk-to-LUN Mappings	Host Does Not Have Individual Virtual Disk-to-LUN Mappings
	under the new host group defined by *hostGroupName*.	under the new host group defined by *hostGroupName*.
none	The host is removed from the host group as an independent partition and is placed under the root node.	The host is removed from the present host group and is placed under the default group.
defaultGroup	The command fails.	The host is removed from the present host group and is placed under the default group.

NOTE: When you use this command, you can specify one or more of the optional parameters. You do not, however, need to use all of the parameters.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Set Host Channel

Description

This command defines the loop ID for the host channel.

Syntax

```
set hostChannel [hostChannelNumber]
preferredID=portID
```

Parameters

Parameter	Description
`hostChannel`	The identifier number of the host channel for which you want to set the loop ID. The host channel identifier number is constructed by combining the RAID controller module number and the host channel number. Enclose the host channel identifier number in square brackets ([ ]). Separate the RAID controller module number from the host channel number with a comma. For example, the identifier number for host channel 2 on RAID controller module 0 would be [0,2].
`preferredID`	The port identifier for the specified host port. Port ID values are `0` to `127`.

Set Host Group

Description

This command renames a host group.

Syntax

```
set hostGroup [hostGroupName]
userLabel="newHostGroupName"
```

Parameters

Parameter	Description
`hostGroup`	Name of the host group to rename. You must put brackets ([ ]) around the host group name. If the host group name has special characters, you must also put quotation marks (" ") around the host group name.
`userLabel`	New name for the host group. You must put quotation marks (" ") around the host group name.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Set Host Port

Description

This command changes the host type for an HBA host port. You can also change an HBA host port label with this command.

Syntax

```
set hostPort [portLabel] host="hostName"
userLabel="newPortLabel"
```

Parameters

Parameter	Description
`hostPort`	The name of the HBA host port for which you want to change the host type, or for which you want to create a new name. Enclose the HBA host port name in square brackets ([ ]). If the HBA host port

Parameter	Description
	label has special characters, enclose the HBA host port label in double quotation marks (" ").
host	The name of the host to which the HBA host port is connected. Enclose the host name in double quotation marks (" ").
userLabel	The new name that you want to give to the HBA host port. Enclose the new name of the HBA host port in double quotation marks (" ").

NOTE: When you use this command, you can specify one or more of the optional parameters. You do not, however, need to use all of the parameters.

NOTE: You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Command names can have a maximum of 30 characters. If you exceed the maximum character limit, replace square brackets ([ ]) with angle brackets (< >) to overcome this limitation.

Set iSCSI Initiator

Description

This command sets the attributes for an iSCSI initiator.

Syntax

```
set iscsiInitiator (["iscsiID"] | <"wwID">)
(userLabel="newName" |
host="newHostName" |
chapSecret="newSecurityKey")
```

Parameters

Parameter	Description
iscsiID	The name of the iSCSI initiator for which you want to set attributes.
userLabel	The new name that you want to use for the iSCSI initiator.
host	The name of the new host to which the HBA host port is connected. Enclose the host name in double quotation marks (" ").
chapSecret	The password that you want to use to authenticate a peer connection.

NOTE: Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a “secret.” A secret is a security key that is similar to a password.

NOTE: Use the `chapSecret` parameter only for initiators requiring mutual authentication.

Set iSCSI Target Properties

Description

This command defines properties for an iSCSI target.

Syntax

```
set iscsiTarget ["iscsiID"]
(authenticationMethod=(none | chap) |
chapSecret="securityKey" |
targetAlias="userLabel")
```

Parameters

Parameter	Description
`iscsiTarget`	The iSCSI target for which you want to set properties. Enclose the name in double quotation marks (" "). You must also enclose the iscsi ID in either square brackets ([ ]) or angle brackets (< >).
`authenticationMethod`	The means of authenticating your iSCSI session.
`chapSecret`	The password that you want to use to authenticate a peer connection.
`targetAlias`	The name that you want to use for the target.

NOTE: Challenge Handshake Authentication Protocol (CHAP) is a protocol that authenticates the peer of a connection. CHAP is based upon the peers sharing a "secret." A secret is a security key that is similar to a password.

NOTE: Use the `chapSecret` parameter only for initiators requiring mutual authentication.

NOTE: The `targetAlias` is a descriptive name that you can give to the target to help make it easier to identify. You can use any combination of alphanumeric characters, hyphens, and underscores for the `targetAlias`. The `targetAlias` can have a maximum of 30 characters.

Set Physical Disk Channel Status

Description

This command defines how the physical disk channel performs.

Syntax

```
set physicalDiskChannel [(1 | 2)]
status=(optimal | degraded)
```

Parameters

Parameter	Description
`physicalDiskChannel`	Identifier number of the physical disk channel for which to set the status. Valid physical disk channel values are `1` or `2`. You must put brackets ([ ]) around the physical disk channel number.
`status`	Condition of the channel. You can set the channel status to `optimal` or `degraded`. NOTE: Use the `optimal` parameter to move a degraded channel back to the Optimal state. Use the `degraded` parameter if the channel is experiencing problems, and the storage array requires additional time for data transfers.

Set Physical Disk Hot Spare

Description

This command assigns or unassigns one or more physical disks as a hot spare.

Syntax

```
set (physicalDisk [enclosureID,slotID] |
physicalDisks [enclosureID1,slotID1 enclosureID2, slotID2...
enclosureIDn,slotIDn])
hotSpare=(TRUE | FALSE)
```

Parameters

Parameter	Description
`physicalDisk or physicalDisks`	The enclosure and the slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. Enclose the enclosure ID values and the slot ID values in square brackets ([ ]).
`hotSpare`	The setting to assign the physical disk as the hot spare. To assign the physical disk as the hot spare, set this parameter to `TRUE`. To remove a hot spare assignment from a physical disk, set this parameter to `FALSE`.

Set Physical Disk State

Description

This command sets a physical disk to the Failed state. To return a physical disk to the Optimal state, use the revive physicalDisk command. Optionally, you can copy the data from the physical disk before you fail it.

Syntax

```
set physicalDisk [enclosureID,drawerID,slotID]
operationalState=failed
[copyPhysicalDisk]
```

Parameters

Parameter	Description
`physicalDisk or physicalDisks`	The enclosure and the slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. Enclose the enclosure ID values and the slot ID values in square brackets ([ ]).
`copyPhysicalDisk`	This parameter copies the data from the physical disk that you are replacing to the replacement physical disk.

Set RAID Controller Module

Description

This command defines the attributes for the RAID controller modules.

Syntax

```
set controller [(0 | 1)]
[availability=(online | offline | serviceMode)]
[ethernetPort [(0 | 1)] ethernetPortOptions]
[globalNVSRAMByte [nvsramOffset]=(nvsramByteSetting | nvsramBitSetting)]
[hostNVSRAMByte [hostType, nvsramOffset]=(nvsramByteSetting |
nvsramBitSetting)]
[IPv4GatewayIP=ipAddress]
[IPv6RouterAddress=ipv6Address]
[iscsiHostPort [(0 | 1 | 2 | 3)] iscsiHostPortOptions]
[rloginEnabled=(TRUE | FALSE)]
[serviceAllowedIndicator=(on | off)]
```

Parameters

Parameter	Description
`controller`	The RAID controller module for which you want to define properties. Valid RAID controller module identifiers are **0** or **1**, where **0** is the RAID controller module in the top slot, and **1** is the RAID controller module in the bottom slot. Enclose the identifier for the RAID controller module in square brackets ([ ]). If you do not specify a RAID controller module, the firmware for the controller returns a syntax error.
`availability`	The mode for the RAID controller module, which you can set to `online`, `offline`, or `serviceMode` (service).
`ethernetPort`	The attributes (options) for the management Ethernet ports. The entries to support this parameter are listed in the Syntax Element Statement Data table that follows. Many settings are possible, including setting the IP address, the gateway address, and the subnet mask address.
`globalNVSRAMByte`	A portion of the RAID controller module NVSRAM. Specify the region to be modified using the starting byte offset within the region and the byte value or bit value of the new data to be stored into the NVSRAM.
`hostNVSRAMByte`	The NVSRAM for the host-specific region. The setting specifies the host index for the specific host, the starting offset within the region, the number of bytes, and the byte value or bit value of the new data to be stored into the NVSRAM.
`IPv4GatewayIP`	The IP address of the node that provides the interface to the network. The address format for the IPv4 gateway is **(0–255).(0–255).(0–255).(0–255)**
`IPv6RouterAddress`	The IP address of IPv6 router that connects two or more logical subnets. The address format for the IPv6 router is **(0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF): (0–FFFF):(0–FFFF):(0–FFFF):(0–FFFF)**.
`iscsiHostPort`	This parameter enables you to set options for the iSCSI ports on the RAID controller module. Select the specific iSCSI port and then select the options for that port. The iSCSI port identifier can be **0**, **1** or **2**, **3**. Refer to the information about your RAID controller module for the location and identifier of the ports.

Parameter	Description
	The values that support this parameter are listed in the Syntax Element Statement Data table that follows. Many settings are possible, including setting the IP address, the gateway address, the subnet mask address, the IPv4 priority, and the IPv6 priority.
`rloginEnabled`	The setting for whether the remote login feature is turned on or turned off. To turn on the remote login feature, set this parameter to **`TRUE`**. To turn off the remote login feature, set this parameter to **`FALSE`**.
`serviceAllowedIndicator`	The setting for whether the Service Action Allowed indicator light is turned on or turned off. To turn on the Service Action Allowed indicator light, set this parameter to **`on`**. To turn off the Service Action Allowed indicator light, set this parameter to **`off`**.

Syntax Element Statement Data

Options for the `Ethernet Port` Parameter

```
enableIPv4=(TRUE | FALSE) |
enableIPv6=(TRUE | FALSE) |
IPv6LocalAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-
FFFF):(0-FFFF) |
IPv6RoutableAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-
FFFF):(0-FFFF) |
IPv4Address=(0-255).(0-255).(0-255).(0-255) |
IPv4ConfigurationMethod=[(static | dhcp)] |
IPv4SubnetMask=(0-255).(0-255).(0-255).(0-255) |
duplexMode=(TRUE | FALSE) |
portSpeed=[(autoNegotiate | 10 | 100 | 1000)]
```

Options for the `iscsiHostPort` Parameter

```
IPv4Address=(0-255).(0-255).(0-255).(0-255) |
IPv6LocalAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-
FFFF):(0-FFFF) |

IPv6RoutableAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-
FFFF):(0-FFFF) |
IPv6RouterAddress=(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-
FFFF):(0-FFFF) |
enableIPv4=(TRUE | FALSE) |
enableIPv6=(TRUE | FALSE) |
enableIPv4Vlan=(TRUE | FALSE) |
enableIPv6Vlan=(TRUE | FALSE) |
enableIPv4Priority=(TRUE | FALSE) |
enableIPv6Priority=(TRUE | FALSE) |
IPv4ConfigurationMethod=(static | dhcp) |
IPv6ConfigurationMethod=(static | auto) |
IPv4GatewayIP=(TRUE | FALSE) |
IPv6HopLimit=[0-255] |
IPv6NdDetectDuplicateAddress=[0-256] |
IPv6NdReachableTime=[0-65535] |
```

Options for the `Ethernet Port` Parameter

```
IPv6NdRetransmitTime=[0-65535] |
IPv6NdTimeOut=[0-65535] |
IPv4Priority=[0-7] |
IPv6Priority=[0-7] |
IPv4SubnetMask=(0-255).(0-255).(0-255).(0-255) |
IPv4VlanId=[1-4094] |
IPv6VlanId=[1-4094] |
maxFramePayload=[frameSize] |
tcpListeningPort=[3260, 49152-65536] |
portSpeed=[(autoNegotiate | 1 | 10)]
```

Notes

When you use this command, you can specify one or more parameters. You need not to use all the parameters.

Setting the availability parameter to `serviceMode` causes the alternate RAID controller module to take ownership of all of the virtual disks. The specified RAID controller module no longer has any virtual disks and refuses to take ownership of any more virtual disks. Service mode is persistent across reset cycles and power cycles until the availability parameter is set to `online`.

Use the `show controller NVSRAM` command to show the NVSRAM information. Before making any changes to the NVSRAM, contact your Technical Support Representative to learn what regions of the NVSRAM you can modify.

When the `duplexMode` option is set to `TRUE`, the selected Ethernet port is set to full duplex. The default value is half duplex (the `duplexMode` parameter is set to `FALSE`).

To make sure that the IPv4 settings or the IPv6 settings are applied, you must set these `iscsiHostPort` options:

- `enableIPV4=TRUE`
- `enableIPV6=TRUE`

The IPv6 address space is 128 bits. It is represented by eight 16-bit hexadecimal blocks separated by colons.

The `maxFramePayload` option is shared between IPv4 and IPv6. The payload portion of a standard Ethernet frame is set to `1500`, and a jumbo Ethernet frame is set to `9000`. When using jumbo frames, all of the devices that are in the network path should be capable of handling the larger frame size.

The `portSpeed` option is expressed as megabits per second (Mb/s).

Values for the `portSpeed` option of the iscsiHostPort parameter are in megabits per second (Mb/s).

The following values are the default values for the **`iscsiHostOptions`**:

- The `IPv6HopLimit` option is 64.
- The `IPv6NdReachableTime` option is 30000 milliseconds.
- The `IPv6NdRetransmitTime` option is 1000 milliseconds.
- The `IPv6NdTimeOut` option is 30000 milliseconds.

- The `tcpListeningPort` option is 3260.

Set Read-Only Snapshot Virtual Disk To A Read/Write Virtual Disk

Description

This command changes a snapshot virtual disk that is a read-only virtual disk to a snapshot virtual disk that is read/write virtual disk. You can also use this command to define the repository for the read/write virtual disk.

Syntax

```
set snapVirtualDisk ["snapImageVirtualDiskName"] convertReadOnlyToReadWrite
(repositoryVirtualDisk="repos_xxxx" | repositoryFullLimit=percentValue)
```

Parameters

Parameter	Description
`snapVirtual Disk`	The name of the snapshot virtual disk that you want to change from read-only to read/write. Enclose the snapshot virtual disk identifier in double quotation marks (" ") inside of square brackets ([ ]).
`repositoryVirtualDisk`	The name of the repository virtual disk has the snapshot image. All repository identifiers have this form: `repos_xxxx` where `xxxx` represents a four digit numerical value. Enclose the snapshot virtual disk name in double quotation marks (" ").
`repositoryFullLimit`	The percentage of repository virtual disk capacity at which you receive a warning that the snapshot image repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.

Notes

The repository virtual disk name is automatically created by the MD storage management software and the firmware when you create a new snapshot group. You cannot rename the repository virtual disk because renaming the repository virtual disk will break the linkage with the snapshot images.

Set Remote Replication

Description

NOTE: This command is valid only with Fibre Channel storage arrays.

This command defines the properties for a remote replication pair.

NOTE: Starting with firmware 8.10, some RAID controller modules no longer support the `writeOrder` parameter and the `writeMode` parameter.

Syntax

```
set remoteReplication (localVirtualDisk [virtualDiskName] |
localVirtualDisks [virtualDiskName1 ... virtualDiskNameN])
[role=(primary | secondary)]
[syncPriority=(highest | high | medium | low | lowest)]
[autoResync=(enabled | disabled)]
[writeOrder=(preserved | notPreserved)]
[writeMode=(synchronous | asynchronous)] [force=(TRUE | FALSE)]
```

Parameters

Parameter	Description
localVirtualDisk	The name local virtual disk for which you want to define properties. Enclose the primary virtual disk name in square brackets ([ ]). If the primary virtual disk name has special characters or numbers, you must enclose the primary virtual disk name in double quotation marks (" ") inside square brackets.
localVirtualDisks	The names of several local virtual disks for which you want to define properties. Enter the names virtual disks using these rules: • Enclose all names in square brackets ([ ]). • Separate each names with a space. If the virtual disk names have special characters or consist only of numbers, enter the names using these rules: • Enclose all names in square brackets ([ ]). • Enclose each names in double quotation marks (" "). • Separate each names with a space.
role	The setting for the local virtual disk to act as the primary virtual disk or the secondary virtual disk. To define the virtual disk as the primary virtual disk, set this parameter to **`primary`**. To define the virtual disk as the secondary virtual disk, set this parameter to **`secondary`**. This parameter applies only when the virtual disk is part of a replication relationship.
syncPriority	The priority that full synchronization has relative to host I/O activity. Valid values are **`highest`**, **`high`**, **`medium`**, **`low`**, or **`lowest`**.
autoResync	The settings for automatic `resynchronization` between the primary virtual disks and the

Parameter	Description
	secondary virtual disks of a remote-replicated pair. This parameter has these values: • **enabled** – Automatic `resynchronization` is turned on. You do not need to do anything further to `resynchronize` the primary virtual disk and the secondary virtual disk. • **disabled** – Automatic `resynchronization` is turned off. To `resynchronize` the primary virtual disks and the secondary virtual disk, you must run the **resume remoteReplication** command.
writeOrder	This parameter defines write order for data transmission between the primary virtual disk and the secondary virtual disk. Valid values are **preserved** or **notPreserved**.
writeMode	This parameter defines how the primary virtual disk writes to the secondary virtual disk. Valid values are **synchronous** or **asynchronous**.
force	The role reversal is forced if the communications link between the storage arrays is down and promotion or demotion on the local side results in a dual-primary condition or a dual-secondary condition. To force a role reversal, set this parameter to **TRUE**. The default value is **FALSE**. NOTE: Use the force parameter only when using the role parameter.

Notes

When you use this command, you can specify one or more optional parameters.

Synchronization priority defines the amount of system resources that are used to synchronize the data between the primary virtual disks and the secondary virtual disks of a replication relationship. If you select the highest priority level, the data synchronization uses the most system resources to perform the full synchronization, which decreases the performance for host data transfers.

The `writeOrder` parameter applies only to asynchronous write modes and makes them become part of a consistency group. Setting the `writeOrder` parameter to **preserved** causes the remote-replicated pair to transmit data from the primary virtual disk to the secondary virtual disk in the same order as the host writes to the primary virtual disk. In the event of a transmission link failure, the data is buffered until a full synchronization can occur. This action can require additional system overhead to maintain the buffered data, which slows operations. Setting the `writeOrder` parameter to **notPreserved** frees the system from having to maintain data in a buffer, but it requires forcing a full synchronization to make sure that the secondary virtual disk has the same data as the primary virtual disk.

Set Remote Replication Group

Description

Use this command to change the synchronization settings and warning thresholds that the RAID controller module owner of the primary side of the remote replication group uses when it performs an initial synchronization or resynchronization. Changing the synchronization settings affects the synchronization operations of all replicated pairs within the remote replication group.

Syntax

```
set asyncRemoteReplicationGroup [asyncRemoteReplicationGroupName]
[virtualDisk="repos_xxxx" increaseRepositoryCapacity
(repositoryVirtualDisk=("repos_xxxx")] [syncInterval=integer (minutes | hours |
days)] [warningSyncThreshold=integer (minutes | hours | days)]
[warningRecoveryThreshold=integer (minutes | hours | days)]
[userLabel="New_asyncReplicationGroupName"]
[warningThresholdPercent=percentValue] [role=(primary | secondary)]
[autoResync=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group that you want to modify. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ") inside square brackets. If the virtual disk name consists only of numbers, such as "1002," you also must enclose the virtual disk name in double quotation marks (" ") inside square brackets.
`syncInterval`	Specify the length of time between automatically sending updates of modified data from the local storage array to the remote storage array. You can specify the length of time in minutes, hours, or days.
`warningSyncThreshold`	Specify the length of time to wait until a warning is triggered for cases in which the synchronization of all of the virtual disks within the remote replication group takes longer than the defined time. You can specify the length of time in minutes, hours, or days.
`warningRecoveryThreshold`	Specify the length of time to wait until a warning is triggered when the automatic data update for the point-in-time image on the remote storage array is older than the defined time. Define the threshold from the end of the previous update. You can

Parameter	Description
	specify the length of time in minutes, hours, or days. NOTE: You must set the Recovery Point Threshold to be twice as large as the synchronization interval threshold.
`warningThresholdPercent`	Specify the percent value that determines when a warning is triggered when the capacity of a replication repository virtual disk reaches the defined percentage. Define the threshold by percentage (%) of the capacity remaining.
`userLabel`	The new name that you want to give to the remote replication group. Use this parameter when you want to rename the remote replication group. Enclose the new remote replication group name in double quotation marks (" ").
`autoResync`	The settings for automatic resynchronization between the primary virtual disks and the secondary virtual disks of a remote replicated pair within a remote replication group. This parameter has these values: • `enabled`—Automatic resynchronization is turned on. You do not need to do anything further to resynchronize the primary virtual disk and the secondary virtual disk. • `disabled`—Automatic resynchronization is turned off. To resynchronize the primary virtual disk and the secondary virtual disk, you must run the `resume remoteReplicationGroup` command.
`virtualDisk`	The name of the specific virtual disk that you want to modify in the remote replication group. Enclose the virtual disk name in double quotation marks (" ").
`repositoryVirtualDisk`	The name of the repository virtual disk for which you want to increase capacity. An available standard virtual disk is added to the repository virtual disk to increase the capacity of the repository virtual disk. You have two options for defining the name of a repository virtual disk: • Use an existing repository virtual disk name. • Create a new repository virtual disk when you run this command. The name of an existing repository virtual disk is comprised of two parts: • The term `repos`

Parameter	Description
	• A four digit numerical identifier that you assign to the repository virtual disk name. Enclose the name of the existing repository virtual disk in double quotation marks (" "). If you want to create a new repository virtual disk when you run this command you must enter the name of either a a disk group or a disk pool in which you want the repository virtual disk. Optionally, you also can define the capacity of the repository virtual disk. If you want to define the capacity you can use these values: • An integer value that represents a percentage of the base virtual disk capacity. • A decimal fraction value that represents a percentage of the base virtual disk capacity. • A specific size for the repository virtual disk. Size is defined in units of `bytes`, `KB`, `MB`, `GB`, or `TB`. If you do not use the capacity option, the MD storage management software sets the capacity to 20 percent of the base virtual disk capacity. Enclose the name of the new repository virtual disk in parenthesis ( ).
`role`	Use this parameter to promote the remote replication group to a primary role or demote the remote replication group to a secondary role. To define the remote replication group as the primary role, set this parameter to `primary`. To define the remote replication group as the secondary role, set this parameter to `secondary`.
`force`	The role reversal is forced if the communications link between the storage arrays is down and promotion or demotion on the local side results in a dual-primary condition or a dual-secondary condition. To force a role reversal, set this parameter to `TRUE`. The default value is `FALSE`. **NOTE:** Use the force parameter only when using the role parameter.
`nosync`	This parameter defines whether to perform an initial synchronization between the primary remote replication group and the secondary remote replication group before a role reversal operation is performed. To perform an initial synchronization, set this parameter to `TRUE`. The default value is `FALSE`.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

A remote replication group repository virtual disk is an expandable virtual disk that is structured as a concatenated collection of up to 16 standard virtual disk entities. Initially, an expandable repository virtual disk has only a single element. The capacity of the expandable repository virtual disk is exactly that of the single element. You can increase the capacity of an expandable repository virtual disk by attaching additional standard virtual disks to it. The composite expandable repository virtual disk capacity then becomes the sum of the capacities of all of the concatenated standard virtual disks.

A remote replication group repository virtual disk must satisfy a minimum capacity requirement that is the sum of the following:

- 32 MB to support fixed overhead for the remote replication group and for copy-on-write processing.
- Capacity for rollback processing, which is 1/5000th of the capacity of the base virtual disk.

The minimum capacity is enforced by the RAID controller module firmware and the MD storage management software.

Set Session

Description

This command defines how you want the current script engine session to run.

Syntax

```
set session errorAction=(stop | continue)
password="storageArrayPassword"
performanceMonitorInterval=intervalValue
performanceMonitorIterations=iterationValue
```

Parameters

Parameter	Description
`errorAction`	How the session responds if an error is encountered during processing. You can choose to stop the session if an error is encountered, or you can continue the session after encountering an error. The default value is stop. (This parameter defines the action for execution errors, not syntax errors. Some error conditions might override the **continue** value.)
`password`	The password for the storage array. Enclose the password in double quotation marks (" ").

Parameter	Description
`performanceMonitorInterval`	The frequency of gathering performance data. Enter an integer value for the polling interval, in seconds, for which you want to capture data. The range of values is 3 to 3600 seconds. The default value is 5 seconds.
`performanceMonitorIterations`	The number of samples to capture. Enter an integer value. The range of values for samples captured is 1 to 3600. The default value is 5.

Additional Information

When you use this command, you can specify one or more of the optional parameters.

Passwords are stored on each storage array in a management domain. If a password was not previously set, you do not need a password. The password can be any combination of alphanumeric characters with a maximum of 30 characters. (You can define a storage array password by using the `set storageArray` command.)

The polling interval and the number of iterations that you specify remain in effect until you end the session. After you end the session, the polling interval and the number of iterations return to their default values.

Set Snapshot Group Attributes

Description

This command defines the properties for a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"] (userLabel="snapGroupName" |
repositoryFullPolicy=(failBaseWrites | purgeSnapImages) |
repositoryFullLimit=percentValue |autoDeleteLimit=numberOfSnapImages |
rollbackPriority=(lowest | low | medium | high | highest))
```

Parameters

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group for which you are setting properties. Enclose the snapshot group name in double quotation marks (" ") inside of square brackets ([ ]).
`userLabel`	The new alphanumeric identifier (including - and _) that you want to give to the snapshot group. Use this parameter when you want to rename the snapshot group. Enclose the new snapshot group name in double quotation marks (" ").

Parameter	Description
repositoryFullPolicy	How you want snapshot image processing to continue if the repository virtual disks for the snapshot image group are full. You can choose to fail writes to the base virtual disk (failBaseWrites) or delete (purge) the snapshot images (purgeSnapImages). The default action is purgeSnapImages.
repositoryFullLimit	The percentage of repository virtual disk capacity at which you receive a warning that the snapshot image repository virtual disk is nearing full. Use integer values. For example, a value of 70 means 70 percent. The default value is 75.
autoDeleteLimit	The maximum number of snapshot images that you want to automatically delete if you have selected to purge the snapshot images for a repository full policy. Use integer values. The default value is 32.
rollBackPriority	Use this parameter to determine whether system resources should be allocated to the rollback operation at the expense of system performance. Valid values are highest, high, medium, low, or lowest. A value of high indicates that the rollback operation is prioritized over all other host I/O. A value of lowest indicates that the rollback operation should be performed with minimal impact to host I/O.

Notes

You can use any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#) for the names. Names can have a maximum of 30 characters.

When you use this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

Set Snapshot Group Media Scan

Description

This command runs a media scan on a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"]
mediaScanEnabled=(TRUE | FALSE)
consistencyCheckEnabled=(TRUE | FALSE)
```

Parameters

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group on which you want to run a media scan. Enclose the snapshot group name in double quotation marks (" ") inside of square brackets ([ ]).
`mediaScanEnabled`	The setting to turn on or turn off media scan for the virtual disk. To turn on media scan, set this parameter to `TRUE`. To turn off media scan, set this parameter to `FALSE`. (If media scan is disabled at the storage array level, this parameter has no effect.)
`consistencyCheckEnabled`	The setting to turn on or turn off consistency checking during a media scan. To turn on consistency checking, set this parameter to `TRUE`. To turn off consistency checking, set this parameter to `FALSE`.

Set Snapshot Group Repository Virtual Disk Capacity

Description

This command increases or decreases the capacity of a snapshot group repository virtual disk.

Syntax For Increasing Capacity

```
set snapGroup ["snapGroupName"]
increaseRepositoryCapacity repositoryVirtualDisks=(repos_xxxx)
```

Syntax For Decreasing Capacity

```
set snapGroup ["snapGroupName"]
decreaseRepositoryCapacity count=numberOfVirtual disks
```

Parameters

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group on which you want to run a media scan. Enclose the snapshot group name in double quotation marks (" ") inside of square brackets ([ ]).
`repositoryVirtualDisk`	The name of the repository virtual disk for which you want to increase capacity. An available standard virtual disk is added to the repository

Parameter	Description
	virtual disk to increase the capacity of the repository virtual disk. The repository virtual disk name is comprised of two parts: • The term `repos` • A numerical identifier generated by the MD storage management software and the firmware
`count`	The number of repository virtual disks that you want to remove from the snapshot group. Use integer values.

Set Snapshot Group Schedule

Description

This command defines the schedule for taking snapshot images for a snapshot group.

Syntax

```
set snapGroup ["snapGroupName"]
enableSchedule=(TRUE | FALSE)
schedule=(immediate | snapshotSchedule)
```

Parameters

Parameter	Description
`snapGroup`	The alphanumeric identifier (including - and _) of the snapshot group for which you are setting properties. Enclose the snapshot group name in double quotation marks (" ") inside of square brackets ([ ]).
`enableSchedule`	Use this parameter to turn on or to turn off the ability to schedule a snapshot operation. To turn on snapshot scheduling, set this parameter to `TRUE`. To turn off snapshot scheduling, set this parameter to `FALSE`.
`schedule`	Use this parameter to schedule a snapshot group operation. You can use one of these options for setting a schedule for a snapshot group operation: • `immediate` • `startDate` • `scheduleDay` • `startTime`

Parameter	Description
	• scheduleInterval • endDate • noEndDate • timesPerDay • timeZone

NOTE: For more information on how to schedule snapshots, see Scheduling Snapshots

Set Snapshot (Legacy) Virtual Disk

Description

This command defines the properties for a snapshot virtual disk and lets you rename a snapshot virtual disk.

Syntax

```
set (virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ...
virtualDiskNameN])
userLabel="snapshotVirtualDiskName"
warningThresholdPercent=percentValue
repositoryFullPolicy=(failSourceWrites |
failSnapshot)
enableSchedule=(TRUE | FALSE) |
schedule=(immediate | snapshotSchedule)
rollbackPriority=(0 | 1 | 2 | 3 | 4)
```

Parameters

Parameter	Description
virtualDisk or virtualDisks	The name of the specific snapshot virtual disk for which you want to define properties. (You can enter more than one virtual disk name if you use the virtualDisks parameter). Enclose the snapshot virtual disk name in double quotation marks (" ") inside of square brackets ([ ]).
userLabel	A new name that you want to give to a snapshot virtual disk. Enclose the new snapshot virtual disk name in double quotation marks (" ").
warningThresholdPercent	The percentage of repository capacity at which you receive a warning that the snapshot repository is nearing full. Use integer values. For example, a value of 70 means 70 percent. The default value is 50.
repositoryFullPolicy	How you want snapshot processing to continue if the snapshot repository virtual disk is full. You can

Parameter	Description
	choose to fail writes to the source virtual disk (`failSourceWrites`) or fail writes to the snapshot virtual disk (`failSnapshot`). The default value is `failSnapshot`.
`enableSchedule`	Turns on or off the snapshot scheduling feature. To turn on snapshot scheduling, set this parameter to `TRUE`. To turn off snapshot scheduling, set this parameter to `FALSE`.
`schedule`	Use this parameter to schedule a snapshot operation, with the following options: • `immediate` • `startDate` • `scheduleDay` • `startTime` • `scheduleInterval` • `endDate` • `noEndDate` • `timesPerDay` • `timeZone` For more information, see Scheduling Snapshots.
`rollbackPriority`	Determines whether system resources must be allocated to the rollback operation at the expense of system performance. A value of 0 indicates that the rollback operation is prioritized over all other host I/O. A value of 4 indicates that the rollback operation must be performed with minimal impact to the host I/O.

Notes

When you use this command, you can specify one or more of the optional parameters. You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

You can set the `warningThresholdPercent` parameter and the `repositoryFullPolicy` parameter for both the snapshot repository virtual disk or the snapshot virtual disk.

Scheduling Snapshots (Legacy)

You can use the `enableSchedule` and the `schedule` parameters to create snapshots for a snapshot group. Automatic snapshots can be scheduled daily, weekly, or monthly (by day or by date). The `enableSchedule` parameter turns on or off the ability to schedule snapshots. When you enable scheduling, you use the `schedule` parameter to define when you want the snapshots to occur.

Parameters

Parameter	Description
immediate	Begins a snapshot operation immediately; a snapshot virtual disk is created and a copy-on-write operation begins.
startDate	Defines a specific start date snapshot virtual disk creation. The format for this option is MM:DD:YY. If no start date is provided, the current date is used. For example: startDate=06:27:11.
scheduleDay	Defines a day of the week you want to create a snapshot virtual disk and perform a copy-on-write operation. The values that you can enter are: monday, tuesday, wednesday, thursday, friday, saturday, sunday, and all. For example: scheduleDay=wednesday.
startTime	Defines time of a day you want to create a snapshot virtual disk and start performing a copy-on-write operation. The format for entering the time is HH:MM, where HH is the hour and MM is the minute past the hour. Use a 24-hour clock. For example: startTime=14:27, which corresponds to 2:27 PM.
scheduleInterval	Amount of time, in minutes that defines a minimum between two copy-on-write operations. It is possible for you to create a schedule in which you have overlapping copy-on-write operations because of the duration of a copy operation. You can ensure that copy-on-write operations maintain a correct interval by using this option. The maximum value for the scheduleInterval option is 1440 minutes. For example: scheduleInterval=180.
endDate	Specific date to stop creating a snapshot virtual disk and end the copy-on-write operations. The format is MM:DD:YY, where MM is the month, DD the day and YY the year. For example: endDate=11:26:11.
noEndDate	Use this option to prevent a scheduled copy-on-write operation from ending. If you later decide to end the copy-on-write operations, you must re-enter the create snapshotVirtualDisk command and specify an end date.
timesPerDay	Number of times you want the snapshot schedule to run in a day. For example, timesPerDay=4.

Parameter	Description
timeZone	Specifies the time zone to use for the scheduling operations, using one of the following values: • GMT±HH:MM—The offset from Greenwich Mean Time (GMT), in hours and minutes. For example, GMT-06:00 must be used to specify Central Standard Time (CST). • Text string—A standard time zone text string, in quotation marks. For example, "USA/Chicago" or "Germany/Berlin". Time zone text strings are case sensitive. If you enter an incorrect text string, the default time zone is GMT.

Examples of code string for defining a schedule:

```
enableSchedule=true schedule startTime=14:27

enableSchedule=true schedule scheduleInterval=180

enableSchedule=true schedule timeZone=GMT-06:00

enableSchedule=true schedule timeZone="USA/Chicago"
```

NOTE: If you also use the scheduleInterval option, the firmware chooses between the timesPerDay option and the scheduleInterval option by selecting the lowest value of the two options. The firmware calculates an integer value for the scheduleInterval option by dividing 1440 by the scheduleInterval option value that you set. For example, 1440/180 = 8. The firmware then compares the timesPerDay integer value with the calculated scheduleInterval integer value and uses the smaller value.

NOTE: To remove a schedule, use the delete virtualDisk command with the schedule parameter. The delete virtualDisk command with the schedule parameter deletes only the schedule, not the snapshot virtual disk.

Set Snapshot Virtual Disk Media Scan

Description

This command runs a media scan on the physical disks used for a snapshot virtual disk. You also can perform a consistency check on the data.

Syntax

```
set snapVirtualDisk ["snapVirtualDiskName"]
mediaScanEnabled=(TRUE | FALSE)
[consistencyCheckEnabled=(TRUE | FALSE)]
```

Parameters

Parameter	Description
snapVirtualDisk	The alphanumeric identifier (including - and _) of the snapshot virtual disk for which you are setting

Parameter	Description
	properties. Enclose the snapshot virtual disk identifier in double quotation marks (" ") inside of square brackets ([ ]).
`mediaScanEnabled`	The setting to turn on or turn off media scan for the snapshot virtual disk. To turn on media scan, set this parameter to `TRUE`. To turn off media scan, set this parameter to `FALSE`. (If media scan is disabled at the storage array level, this parameter has no effect.)
`consistencyCheckEnabled`	The setting to turn on or turn off consistency checking during a media scan. To turn on consistency checking, set this parameter to `TRUE`. To turn off consistency checking, set this parameter to `FALSE`.

Set Snapshot Virtual Disk Repository Virtual Disk Capacity

Description

This command defines the capacity the you can set for a snapshot virtual disk repository virtual disk.

Syntax

```
set snapVirtualDisk ["snapVirtualDiskName"]
(repositoryFullLimit=percentValue |
increaseRepositoryCapacity (repositoryVirtualDisk=(repos_xxxx) |
decreaseRepositoryCapacity count=numberOfVirtualDisks)
```

Parameters

Parameters	Description
`snapVirtualDisk`	The alphanumeric identifier (including - and _) of the snapshot virtual disk for which you are setting properties. Enclose the snapshot virtual disk identifier in double quotation marks (" ") inside of square brackets ([ ]).
`repositoryFullLimit`	The percentage of repository virtual disk capacity at which you receive a warning that the snapshot repository virtual disk is nearing full. Use integer values. For example, a value of `70` means 70 percent. The default value is `75`.
`repositoryVirtualDisk`	The alphanumeric identifier of the repository virtual disk has the snapshot image. All repository identifiers have this form: `repos_xxxx` where `xxxx` represents a four digit numerical value.

Parameters	Description
	Enclose the snapshot virtual disk name in double quotation marks (" ").
`count`	The number of repository virtual disks that you want to remove from the snapshot group. Use integer values.

Notes

When you use this command, you can specify only one of the parameters. You cannot enter more than one parameter when you run this command.

Set Storage Array

NOTE: With firmware version 8.10 the cacheFlushStop parameter is deprecated and will be removed in a future release of storage management software.

Description

The means of registering the iSCSI target on the iSNS server. Set the parameter to TRUE to list an iSCSI target. When you set this parameter to TRUE you must also use these parameters for IPV4 configurations:

- `isnsIPV4ConfigurationMethod`
- `isnsIPV4Address`

When you set this parameter to TRUE you must also use these parameters for IPV6 configurations:

- `isnsIPV6Address`

Optionally, you can also use the isnsListeningPort parameter to define the port monitor and manage traffic to the server. To remove the registration for the storage array from the iSNS server set this parameter to FALSE

Syntax

```
set storageArray (cacheBlockSize=cacheBlockSizeValue |
cacheFlushStart=cacheFlushStartSize |
defaultHostType=("hostTypeName" | "hostTypeIdentifier")
failoverAlertDelay=delayValue |
mediaScanRate=(disabled | 1-30) |
userRole=(admin | monitor)
password="password" |
userLabel="storageArrayName"
isnsRegistration=(TRUE | FALSE))
```

Parameters

Parameter	Description
`cacheBlockSize`	The cache block size that is used by the RAID controller module for managing the cache. Valid

Parameter	Description
	values are **4** (4 KB), **8** (8 KB), **16** (16 KB), or **32** (32 KB).
`cacheFlushStart`	The percentage of unwritten data in the cache that causes a cache flush. Use integer values from **0** to **100** to define the percentage. The default value is **80**.
`defaultHostType`	The default host type of any unconfigured host port to which the RAID controller modules are connected. To generate a list of valid host types for the storage array, run the **`show storageArray hostTypeTable`** command. Host types are identified by a name or a numerical index. Enclose the host type name in double quotation marks (" "). Do not enclose the host type numerical identifier in double quotation marks.
`failoverAlertDelay`	The failover alert delay time in minutes. The valid values for the delay time are **0** to **60** minutes. The default value is **5**.
`mediaScanRate`	The number of days over which the media scan runs. Valid values are **`disabled`**, which turns off the media scan, or **1** day to **30** days, where **1** day is the fastest scan rate, and **30** days is the slowest scan rate. A value other than **`disabled`** or **1** to **30** does not allow the media scan to function.
`password`	The password for the storage array. Enclose the password in double quotation marks (" ").
`userRole`	Defines the user role for the password. The user role can be either: • admin – Enables you to view and modify storage array configurations • monitor – Enables you to view storage array configurations and monitor storage array health conditions, but not modify the configuration The `userRole` parameter is optional. If you do not use the `userRole` parameter, the storage array uses any password that you define as the administrator password.
`userLabel`	The name for the storage array. Enclose the storage array name in double quotation marks (" ").
`isnsRegistration`	The means of listing the iSCSI target on the iSNS server. Refer to the set `storageArray isnsRegistration` command for information about how to use this parameter.

Parameter	Description
	NOTE: No other command options can be used in conjunction with `isnsRegistration`.

Notes

When you use this command, you can specify one or more of the optional parameters.

Cache Block Size

When you define cache block sizes, use the 4-KB cache block size for storage arrays that require I/O streams that are typically small and random. Use the 8-KB cache block size when the majority of your I/O streams are larger than 4 KB but smaller than 8 KB. Use the 16-KB cache block size or the 32-KB cache block size for storage arrays that require large data transfer, sequential, or high-bandwidth applications.

The `cacheBlockSize` parameter defines the supported cache block size for all the virtual disks in the storage array. Not all RAID controller module types support all cache block sizes. For redundant configurations, this parameter includes all the virtual disks that are owned by both RAID controller modules within the storage array.

Cache Flush Start and Cache Flush Stop

When you define values to start a cache flush, a value that is too low increases the chance that data needed for a host read is not in the cache. A low value also increases the number of physical disk writes that are necessary to maintain the cache level, which increases system overhead and decreases performance.

When setting storage array cache settings, the value of the `cacheFlushStart` parameter must always be greater than or equal to the value of the `cacheFlushStart` parameter. For example, if the value of the `cacheFlushStart` parameter is set to **80**, you may set the value of the `cacheFlushStart` parameter within the range of **0** to **80**.

When you define values to stop a cache flush, the lower the value, the higher the chance that the data for a host read requires a physical disk read rather than reading from the cache.

Default Host Type

When you define host types, if Storage Partitioning is enabled, the default host type affects only those virtual disks that are mapped in the default group. If Storage Partitioning is not enabled, all of the hosts that are attached to the storage array must run the same operating system and be compatible with the default host type.

Media Scan Rate

Media scan runs on all of the virtual disks in the storage array that have **Optimal** status, do not have modification operations in progress, and have the `mediaScanRate` parameter enabled. Use the **`set virtualDisk`** command to enable or disable the `mediaScanRate` parameter.

Password

Passwords are stored on each storage array. For best protection, the password must meet these criteria:

- The password must be between eight and 30 characters long.
- The password must contain at least one uppercase letter.
- The password must contain at least one lowercase letter.
- The password must contain at least one number.
- The password must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE: If you are using self encrypting physical disks in your storage array, you must use these criteria for your storage array password.

Set Storage Array Enclosure Positions

Description

This command defines the position of the enclosures in a storage array. You must include all of the enclosures in the storage array when you enter this command.

Syntax

```
set storageArray enclosurePositions=(controller | enclosureID... enclosureIDn)
```

Parameter

Parameter	Description
`enclosurePostions`	A list of all of the enclosure IDs. The sequence of the enclosure IDs in the list defines the positions for the RAID enclosure and the expansion enclosures in a storage array. Valid values are `0` to `99`. Enter the enclosure ID values separated by a space. Enclose the list of enclosure ID values in parentheses. For storage arrays where the RAID enclosure has a predefined identifier that is not in the range of valid enclosure position values, use the RAID controller module value.

Additional Information

This command defines the position of an enclosure in a storage array by the position of the enclosure ID in the `enclosurePositions` list. For example, if you have a RAID enclosure with an ID set to `84` and expansion enclosures with IDs set to `1`, `12`, and `50`, the `enclosurePositions` sequence (`84 1 12 50`) places the RAID enclosure in the first position, expansion enclosure 1 in the second position, expansion enclosure 12 in the third position, and expansion enclosure 50 in the fourth position. The `enclosurePositions` sequence (`1 84 50 12`) places the RAID enclosure in the second position, expansion enclosure 1 in the first position, expansion enclosure 50 in the third position, and expansion enclosure 12 in the fourth position.

NOTE: You must include all of the enclosures in the storage array in the list defined by the `enclosurePositions` parameter. If the number of enclosures in the list does not match the total number of enclosures in the storage array, an error message is shown.

Set Storage Array ICMP Response

Description

This command returns the default values for negotiable settings for sessions and connections, which represent the starting point for the storage array for negotiations.

Syntax

```
set storageArray icmpPingResponse=(TRUE | FALSE)
```

Parameter

Parameter	Description
`icmpPingResponse`	This parameter turns on or turns off Echo Request messages. Set the parameter to `TRUE` to turn on Echo Request messages. Set the parameter to `FALSE` to turn off Echo Request messages.

Set Storage Array iSNS Server IPv4 Address

Description

This command sets the configuration method and address for an IPv4 Internet Storage Name Service (iSNS).

Syntax

```
set storageArray isnsIPv4ConfigurationMethod=(static | dhcp)
isnsIPv4Address=ipAddress
```

Parameter

Parameter	Description
`isnsIPv4Address`	The IP address that you want to use for the iSNS server. Use this parameter with the static value for IPv4 configurations. If you choose to have a DHCP server set the IP address for an IPv4 Internet iSNS server, you must set the `isnsIPv4Address` parameter to 0.0.0.0.

Set Storage Array iSNS Server IPv6 Address

Description

This command sets the IPv6 address for the iSNS server.

Syntax

```
set storageArray isnsIPv6Address=ipAddress
```

Parameter

Parameter	Description
`isnsIPv6Address`	The IPv6 address that you want to use for the iSNS server.

Set Storage Array iSNS Server Listening Port

Description

This command sets the iSNS server listening port.

Syntax

```
set storageArray isnsListeningPort=listeningPortIPAddress
```

Parameter

Parameter	Description
`isnsListeningPort`	The IP address that you want to use for the iSNS server listening port. The range of values for the listening port is `49152` to `65535`. The default value is `3205`.

Additional Information

A listening port resides on the database server and is responsible for these activities:

- Listening (monitoring) for incoming client connection requests
- Managing the traffic to the server

When a client requests a network session with a server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

Set Storage Array Learn Cycle

Description

This command sets the learn cycle for the battery backup unit. The learn cycle enables the MD storage management software to predict the remaining battery life. Learn cycles run at set intervals and store the results for software analysis.

Syntax

```
set storageArray learnCycleDate
(daysToNextLearnCycle=numberOfDays |
day=dayOfTheWeek) time=HH:MM
```

Parameters

Parameter	Description
`daysToNextLearnCycle`	Valid values are `0` through `7`, where, `0` is immediately and `7` is in seven days. The `daysToNextLearnCycle` parameter takes place up to seven days after the next scheduled learn cycle.
`day`	Valid values for the day parameter include the days of the week. Setting the day causes the next learn cycle to be scheduled on the specified day, after the currently scheduled learn cycle.
`time`	The time in 24-hour format; for example 8:00 a.m. is entered as 08:00. Nine o'clock p.m. is entered as 21:00, and 9:30 p.m. is entered as 21:30.

NOTE: You can set the learn cycle to occur only once during a seven-day period. The time parameter selects a specific time that you want to run the learn cycle. If a value is not entered, the command uses a default value of 00:00 (midnight). If the day and time specified are in the past, the next learn cycle takes place on the next possible day specified.

Set Storage Array Redundancy Mode

Description

This command sets the redundancy mode of the storage array to either simplex or duplex.

Syntax

```
set storageArray redundancyMode=(simplex | duplex)
```

Parameter

Parameter	Description
`redundancyMode`	Use **simplex** mode when you have a single RAID controller module. Use **duplex** mode when you have two RAID controller modules

Set Storage Array Security Key

Description

Use this command to set the security key that is used throughout the storage array to implement the Physical Disk Security premium feature. When any security-capable physical disk in the storage array is assigned to a secured disk group, that physical disk is security-enabled using the security key. Before you can set the security key, you must use the `create storageArray securityKey` command to create the security key.

Syntax

```
set storageArray securityKey
```

Parameters

None.

Additional Information

Security-capable physical disks have hardware to accelerate cryptographic processing and each has a unique physical disk key. A security-capable physical disk behaves like any other physical disk until it is added to a secured disk group. When a security-capable physical disk is added to a secured disk group, it becomes security-enabled.

Whenever a security-enabled physical disk is powered on, it requires the correct security key from the RAID controller module before it can read or write data. So, a security-enabled physical disk uses two keys: the physical disk key that encrypts and decrypts the data and the security key that authorizes the encryption and decryption processes. The `set storageArray securityKey` command commits the security key to all of the RAID controller modules and security-enabled physical disks in the storage array. The self-encrypting disk feature ensures that if a security-enabled physical disk is physically removed from a storage array, its data cannot be read by any other device unless the security key is known.

Set Storage Array Time

Description

This command sets the clocks on both RAID controller modules in a storage array by synchronizing the RAID controller module clocks with the clock of the host from which you run this command.

Syntax

```
set storageArray time
```

Parameters

None.

Set Storage Array Unnamed Discovery Session

Description

This command enables the storage array to participate in unnamed discovery sessions.

Syntax

```
set storageArray unnamedDiscoverySession=(TRUE | FALSE)
```

Parameter

Parameter	Description
`unnamedDiscoverySession`	This parameter turns on or turns off unnamed discovery sessions. Set the parameter to `TRUE` to turn on unnamed discovery sessions. Set the parameter to `FALSE` to turn off unnamed discovery sessions.

NOTE: Discovery is the process where initiators determine the targets that are available. Discovery occurs at power-on/initialization and also if the bus topology changes, for example, if an extra device is added.

NOTE: An unnamed discovery session is a discovery session that is established without specifying a target ID in the login request. For unnamed discovery sessions, neither the target ID nor the target portal group ID are available to the targets.

Set Thin Virtual Disk Attributes

Description

This command defines the properties for a thin virtual disk. You can use the parameters to define properties for one or more thin virtual disks.

Syntax

```
set (virtualDisk [virtualDiskName] | virtualDisks [virtualDiskName1 ...
virtualDiskNameN] | virtualDisk <"wwID">) [newCapacity=capacityValue]
[repositoryMaxCapacity=capacityValue]  [repositoryExpansionPolicy=(automatic|
manual)]  [warningThresholdPercent=warningThresholdPercentValue]
[addRepositoryCapacity=capacity-spec]
```

Parameters

Parameter	Description
`virtualDisk` or `virtualDisks`	The name of the specific virtual disk for which you want to define properties. (You can enter more than one virtual disk name if you use the `virtualDisks` parameter.) Enclose the virtual disk name in double quotation marks (" ") inside of square brackets ([ ]).
`virtualDisk`	The World Wide Identifier (WWID) of the virtual disk for which you are setting properties. You can use the WWID instead of the virtual disk name to identify the virtual disk. Enclose the WWID in angle brackets (< >).
`newCapacity`	Increase the virtual capacity of the thin virtual disk. This is the value that the virtual disk will report to a host that is mapped to the virtual disk. Values smaller or equal to the existing capacity will cause an error.  **NOTE:** The maximum capacity is 250 GB.
`repositoryMaxCapacity`	Set the maximum capacity of the repository virtual disk. The value must not be smaller that the physical capacity of the repository virtual disk. If the new value results in a reduction in capacity to a level below the warning threshold, the command will produce an error.
`repositoryExpansionPolicy`	Set the expansion policy to `automatic` or `manual`. When you change the policy from `automatic` to `manual`, the maximum capacity value (quota) changes to the physical capacity of the repository virtual disk.
`warningThresholdPercent`	Adjust the warning threshold. If a change to this value would cause a warning condition, the command will fail and produce an error.
`addRepositoryCapacity`	Allocate capacity from the free extent of the disk pool. If insufficient space is available the command fails.

Notes

When you use this command, you can specify one or more of the optional parameters.

Set Virtual Disk

Description

This command defines the properties for a virtual disk. You can use most of the parameters to define properties for one or more virtual disks, however, some of the parameters define properties for only one virtual disk at a time. The syntax definitions are separated to show which parameters apply to several virtual disks and which apply to only one virtual disk. The syntax for virtual disk mapping is listed separately.

Syntax Applicable To One Or More Virtual Disks

```
set (allVirtualDisks | virtualDisk
["virtualDiskName"] |
virtualDisks ["virtualDiskName1" ...
"virtualDiskNameN"] |
virtualDisk <wwID>)
cacheFlushModifier=cacheFlushModifierValue
cacheWithoutBatteryEnabled=(TRUE | FALSE)\
mediaScanEnabled=(TRUE | FALSE)
mirrorCacheEnabled=(TRUE | FALSE)
modificationPriority=(highest | high | medium |
low | lowest) owner=(0 | 1)
readCacheEnabled=(TRUE | FALSE)
writeCacheEnabled=(TRUE | FALSE)
cacheReadPrefetch=(TRUE | FALSE)
```

CAUTION: Enabling *Write Cache* on a virtual disk generally improves performance for applications with significant Write content (unless the application features a continuous string of Writes. However, *Write Cache* does introduce some risk of data loss in the unlikely event of a RAID controller module failure.

Syntax Applicable To Only One Virtual Disk

```
set (virtualDisk ["virtualDiskName"] | virtualDisk <wwID>)
addCapacity=virtualDiskCapacity
[addPhysicalDisks=(enclosureID1,slotID1
enclosureID2,slotID2 ... enclosureIDn,slotIDn)]
consistencyCheckEnabled=(TRUE | FALSE)
segmentSize=segmentSizeValue
userLabel=virtualDiskName
preReadRedundancyCheck=(TRUE | FALSE)
```

Syntax Applicable To Virtual Disk Mapping

```
set (virtualDisk ["virtualDiskName"] | virtualDisk <wwID>)
accessVirtualDisk)
logicalUnitNumber=LUN
(host="hostName" | hostGroup=("hostGroupName" |
defaultGroup)
```

Parameters

Parameter	Description
`allVirtualDisks`	Specifies the properties for all virtual disks in the storage array.
`virtualDisk or virtualDisks (name)`	Specifies the name of the virtual disk for which to define properties. You can enter more than one virtual disk name if you use the `virtualDisks` parameter. You must put quotation marks (" ") and brackets ([ ]) around the virtual disk name. The virtual disk name and quotation marks must be inside the brackets.
`virtualDisk`	Specifies the WWID of the virtual disk for which you are setting properties. You can use the WWID instead of the virtual disk name to identify the virtual disk. You must put angle brackets (< >) around the WWID.  **NOTE:** When running this command, do not use colon separators in the WWID.
`cacheFlushModifier`	The maximum amount of time that data for the virtual disk stays in cache before the data is flushed to physical storage. Valid values are listed in the Notes section.
`cacheWithoutBatteryEnabled`	The setting to turn on or turn off caching without batteries. To turn on caching without batteries, set this parameter to `TRUE`. To turn off caching without batteries, set this parameter to `FALSE`.
`mediaScanEnabled`	The setting to turn on or turn off media scan for the virtual disk. To turn on media scan, set this parameter to `TRUE`. To turn off media scan, set this parameter to `FALSE`. (If media scan is disabled at the storage array level, this parameter has no effect.)
`mirrorCacheEnabled`	The setting to turn on or turn off the replication cache. To turn on the replication cache, set this parameter to `TRUE`. To turn off the replication cache, set this parameter to `FALSE`.
`modificationPriority`	The priority for virtual disk modifications while the storage array is operational. Valid values are `highest`, `high`, `medium`, `low`, or `lowest`.
`owner`	The RAID controller module that owns the virtual disk. Valid RAID controller module identifiers are 0 or 1, where, 0 is the RAID controller module in slot 0, and 1 is the RAID controller module in slot 1. Use

Parameter	Description
	this parameter only if you want to change the virtual disk owner.
readCacheEnabled	The setting to turn on or turn off the read cache. To turn on the read cache, set this parameter to TRUE. To turn off the read cache, set this parameter to FALSE.
writeCacheEnabled	The setting to turn on or turn off the write cache. To turn on the write cache, set this parameter to TRUE. To turn off the write cache, set this parameter to FALSE.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
addCapacity	The setting to increase the storage size (capacity) of the virtual disk for which you are defining properties. Size is defined in units of bytes, KB, MB, GB, or TB. The default value is bytes. NOTE: A space must be added between the last digit and the size (bytes, KB, MB, or GB) for values greater than 9.
addPhysicalDisks	Adds new physical disks to the virtual disk. Specify the enclosure ID and slot ID for each physical disk that you assign to the virtual disk. Enclosure ID values are 0 to 99. Slot ID values are 0 to 31. You must put brackets ([ ]) around the enclosure ID values and the slot ID values. Use with the addCapacity parameter if you must specify additional physical disks to accommodate the new size.
consistencyCheckEnabled	Turns consistency checking during a media scan on or off. To turn consistency checking on, set to TRUE. To turn consistency checking off, set to FALSE.
segmentSize	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512. For information about the segmentSize parameter, see Using The Auto Configure Command.

Parameter	Description
`userLabel`	Specifies the new name to give an existing virtual disk. You must put quotation marks (" ") around the new virtual disk name.
`preReadRedundancyCheck`	The setting to check the consistency of RAID redundancy data on the stripes during read operations. Do not use this operation for nonredundant virtual disks, for example RAID Level 0. To check redundancy consistency, set this parameter to `TRUE`. For no stripe checking, set this parameter to `FALSE`.
`accessVirtualDisk`	Specifies the logical unit number for the access virtual disk. The logical unit number is the only property that you can set for the access virtual disk. **NOTE:** If you specify the `accessVirtualDisk` parameter, the only property that you can set is the `logicalUnitNumber`.
`logicalUnitNumber`	Defines specific virtual disk-to-LUN mappings to an individual host or assigns the host to a host group.
`host`	Specifies the name of the host to which the virtual disk is mapped. You must put quotation marks (" ") around the host name.
`hostGroup`	Specifies the name of the host group to which the virtual disk is mapped. You must put quotation marks (" ") around the host group name.

NOTE: When using this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

NOTE: You can apply the `addCapacity`, `segmentSize`, `userLabel`, and `logicalUnitNumber` parameters to only one virtual disk at a time.

NOTE: Setting the `addCapacity`, `addPhysicalDisks`, or `segmentSize` parameter starts a long-running operation that you cannot stop. These long-running operations are performed in the background and do not prevent you from running other commands. To display the progress of long-running operations, use the `show virtualDisk actionProgress` command.

Set Virtual Disk Attributes For A Disk Pool

Description

This command defines the properties for a virtual disk. You can use most parameters to define properties for one or more virtual disks. You also can use some parameters to define properties for only one virtual

disk. The syntax definitions are separated to show which parameters apply to several virtual disks and which apply to only one virtual disk. Also, the syntax for virtual disk mapping is listed separately.

NOTE: In configurations where disk groups consist of more than 32 virtual disks, the operation can result in host I/O errors or internal RAID controller module reboots due to the expiration of the timeout period before the operation completes. If you experience host I/O errors or internal RAID controller module reboots, bring the host to a quiescent state and try the operation again.

Syntax

```
set (allVirtualDisks | virtualDisk ["virtualDiskName"] | virtualDisks
["virtualDiskName1" ... "virtualDiskNameN"] |
virtualDisk <wwID>)  (addCapacity = capacityValue(KB|MB|GB|TB|Bytes) |
[addPhysicalDisks = (physicalDisk-spec-list)] |
cacheFlushModifier=cacheFlushModifierValue | cacheReadPrefetch = (TRUE | FALSE)
| cacheWithoutBatteryEnabled=(TRUE | FALSE) |
mediaScanEnabled=(TRUE | FALSE) | replicationCacheEnabled=(TRUE | FALSE) |
modificationPriority=(highest | high | medium | low | lowest) |
owner=(0 | 1) | preReadConsistencyCheck = (TRUE | FALSE) |
readCacheEnabled=(TRUE | FALSE) | consistencyCheckEnabled = (TRUE | FALSE) |
segmentSize = segmentSizeValueuserLabel = userlabelValuewriteCacheEnabled=(TRUE
| FALSE) |
dataAssuranceDisabled=(TRUE | FALSE) | flashCacheEnabled=(TRUE | FALSE))
```

Parameters

Parameter	Description
`allVirtualDisks`	The properties for all virtual disks in the storage array.
`virtualDisk` or `virtualDisks`	The name of the specific virtual disk for which you want to define properties. (You can enter more than one virtual disk name if you use the `virtualDisks` parameter.) Enclose the virtual disk name in double quotation marks (" ") inside of square brackets ([ ]).
`virtualDisk`	The World Wide Identifier (WWID) of the virtual disk for which you are setting properties. You can use the WWID instead of the virtual disk name to identify the virtual disk. Enclose the WWID in angle brackets (< >). NOTE: When running this command, do not use colon separators in the WWID.
`addCapacity`	The setting to increase the storage size (capacity) of the virtual disk for which you are defining properties. Size is defined in units of bytes, KB, MB, GB, or TB. The default value is `bytes`. This parameter is not valid for thin virtual disks. NOTE: The maximum capacity is 250 GB.

Parameter	Description
addPhysicalDisks	The setting to add new physical disks to the virtual disk. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value for the physical disk. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value for the physical disk. Enclosure ID values are 0 to 99. Drawer ID values are 1 to 5. Slot ID values are 0 to 31. Enclose the enclosure ID value, drawer ID value, and the slot ID value in parentheses. Use this parameter with the addCapacity parameter if you need to specify additional physical disks to accommodate the new size. This parameter is not valid for disk pool virtual disks or for thin virtual disks. Each disk pool must be created from a minimum of 11 physical disks specified. There is no upward limit of physical disks supported in a single disk pool, but the number cannot exceed the number of physical disks supported on the storage array.
cacheFlushModifier	The maximum amount of time that data for the virtual disk stays in cache before the data is flushed to physical storage. Valid values are listed in the Notes section.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
cacheWithoutBatteryEnabled	The setting to turn on or turn off caching without batteries. To turn on caching without batteries, set this parameter to TRUE. To turn off caching without batteries, set this parameter to FALSE.
mediaScanEnabled	The setting to turn on or turn off media scan for the virtual disk. To turn on media scan, set this parameter to TRUE. To turn off media scan, set this parameter to FALSE. (If media scan is disabled at the storage array level, this parameter has no effect.)
replicationCacheEnabled	The setting to turn on or turn off the replication cache. To turn on the replication cache, set this parameter to TRUE. To turn off the replication cache, set this parameter to FALSE.
modificationPriority	The priority for virtual disk modifications while the storage array is operational. Valid values are highest, high, medium, low, or lowest.

Parameter	Description
preReadConsistencyCheck	The setting to turn on or turn off preread consistency checking. Turning on preread consistency checking verifies the consistency of RAID redundancy data for the stripes containing the read data. Preread consistency checking is performed on read operations only. To turn on preread consistency checking, set this parameter to TRUE. To turn off preread consistency checking, set this parameter to FALSE. NOTE: Do not use this parameter on non redundant virtual disks, such as RAID 0 virtual disks.
readCacheEnabled	The setting to turn on or turn off the read cache. To turn on the read cache, set this parameter to TRUE. To turn off the read cache, set this parameter to FALSE.
consistencyCheckEnabled	The setting to turn on or turn off the write cache. To turn on the write cache, set this parameter to TRUE. To turn off the write cache, set this parameter to FALSE.
segmentSize	The amount of data (in KB) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512
userLabel	The new name that you want to give an existing virtual disk. Enclose the new virtual disk name in double quotation marks (" ").
writeCacheEnabled	The setting to turn on write cache capability.

Notes

When you use this command, you can specify one or more of the optional parameters.

You can apply these parameters to only one virtual disk at a time:

- addCapacity
- segmentSize
- userLabel

Add Capacity, Add Physical Disks, And Segment Size

Setting the addCapacity parameter, the addPhysicalDisks parameter, or the segmentSize parameter starts a long-running operation that you cannot stop. These long-running operations are performed in the background and do not prevent you from running other commands. To show the progress of long-running operations, use the show virtualDisk actionProgress command.

Cache Flush Modifier

Valid values for the cache flush modifier are listed in the following table.

Value	Description
`Immediate`	Data is flushed as soon as it is placed into the cache.
`250`	Data is flushed after 250 ms.
`500`	Data is flushed after 500 ms.
`750`	Data is flushed after 750 ms.
`1`	Data is flushed after 1 s.
`1500`	Data is flushed after 1500 ms.
`2`	Data is flushed after 2 s.
`5`	Data is flushed after 5 s.
`10`	Data is flushed after 10 s.
`20`	Data is flushed after 20 s.
`60`	Data is flushed after 60 s (1 min.).
`120`	Data is flushed after 120 s (2 min.).
`300`	Data is flushed after 300 s (5 min.).
`1200`	Data is flushed after 1200 s (20 min.).
`3600`	Data is flushed after 3600 s (1 hr).
`Infinite`	Data in cache is not subject to any age or time constraints. The data is flushed based on other criteria that are managed by the RAID controller module.

Cache Without Battery Enabled

Write caching without batteries enables write caching to continue if the RAID controller module batteries are completely discharged, not fully charged, or not present. If you set this parameter to `TRUE` without an uninterruptible power supply (UPS) or other backup power source, you can lose data if the power to the storage array fails. This parameter has no effect if write caching is disabled.

Modification Priority

Modification priority defines the amount of system resources that are used when modifying virtual disk properties. If you select the highest priority level, the virtual disk modification uses the most system resources, which decreases the performance for host data transfers.

Cache Read Prefetch

The `cacheReadPrefetch` parameter enables the RAID controller module to copy additional data blocks into cache while the RAID controller module reads and copies data blocks that are requested by the host from the physical disk into cache. This action increases the chance that a future request for data can be fulfilled from cache. Cache read prefetch is important for multimedia applications that use sequential data transfers. The configuration settings for the storage array that you use determine the number of additional data blocks that the RAID controller module reads into cache. Valid values for the `cacheReadPrefetch` parameter are `TRUE` or `FALSE`.

Segment Size

The size of a segment determines how many data blocks that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Each data block stores 512 bytes of data. A data block is the smallest unit of storage. The size of a segment determines how many data blocks that it contains. For example, an 8 KB segment holds 16 data blocks. A 64 KB segment holds 128 data blocks.

When you enter a value for the segment size, the value is checked against the supported values that are provided by the RAID controller module at run time. If the value that you entered is not valid, the RAID controller module returns a list of valid values. Using a single physical disk for a single request leaves other physical disks available to simultaneously service other requests.

If the virtual disk is in an environment where a single user is transferring large units of data (such as multimedia), performance is maximized when a single data transfer request is serviced with a single data stripe. (A data stripe is the segment size that is multiplied by the number of physical disks in the disk group that are used for data transfers.) In this case, multiple physical disks are used for the same request, but each physical disk is accessed only once.

For optimal performance in a multiuser database or file system storage environment, set your segment size to minimize the number of physical disks that are required to satisfy a data transfer request.

Set Virtual Disk Copy

Description

This command defines the properties for a virtual disk copy pair.

Syntax

```
set virtualDiskCopy target [targetName] [source
[sourceName]]
copyPriority=(highest | high | medium | low | lowest)
targetReadOnlyEnabled=(TRUE | FALSE)
copyType=(online | offline)
```

Parameters

Parameter	Description
`target`	Specifies the name of the target virtual disk for which to define properties. You must put brackets ([ ]) around the target virtual disk name. If the target virtual disk name has special characters, you must also put quotation marks (" ") around the target virtual disk name.
`source`	Specifies the name of the source virtual disk for which to define properties. You must put brackets ([ ]) around the source virtual disk name. If the source virtual disk name has special characters, you must also put quotation marks (" ") around the source virtual disk name.
`copyPriority`	Specifies the priority that the virtual disk copy has relative to host I/O activity. Valid entries are `highest`, `high`, `medium`, `low`, or `lowest`.
`targetReadOnlyEnabled`	Specifies whether the target virtual disk is write enabled or read only. To be able to write to the target virtual disk, set to `FALSE`. To prevent writing to the target virtual disk, set to `TRUE`.
`copyType`	Use this parameter to identify that a virtual disk copy has a snapshot (legacy). If the virtual disk copy has a snapshot (legacy), set this parameter to `online`. If the virtual disk copy does not have a snapshot (legacy), set this parameter to `offline`.

NOTE: When using this command, you can specify one or more of the parameters. You do not, however, need to use all of the parameters.

Set Virtual Disk Mapping

Description

This command defines the logical unit number (LUN) mapping between a virtual disk a host or host group. This command is applicable to virtual disks in either a disk group or disk pools.

NOTE: You cannot use this command for a snapshot virtual disk that is used in online virtual disk copy

Syntax

To map a PiT snapshot to the host, use the following syntax:

```
create mapping virtualDisk="snapshotVirtualdiskName" logicalUnitNumber=LUN
host="hostName"
```

For mapping Legacy snapshot and any virtual disk:

```
set (virtualDisk ["virtualDiskName"] | virtualDisk <wwID> | accessVirtualDisk)
logicalUnitNumber=LUN
(host="hostName" |
hostGroup=("hostGroupName" | defaultGroup)
```

Parameters

Parameter	Description
`virtualDisk`	The name of the specific virtual disk for which you want to define mapping. Enclose the virtual disk name in double quotation marks (" ") inside of square brackets ([ ]). The World Wide Identifier (WWID) of the virtual disk for which you are setting properties. You can use the WWID instead of the virtual disk name to identify the virtual disk. Enclose the WWID in angle brackets (< >).
`accessVirtualDisk`	The logical unit number for the virtual disk. The logical unit number is the only property that you can set for the virtual disk.
`logicalUnitNumber`	The logical unit number that you want to use to map to a specific host. This parameter also assigns the host to a host group.
`host`	The name of the host to which the virtual disk is mapped. Enclose the host name in double quotation marks (" ").
`hostGroup`	The name of the host group to which the virtual disk is mapped. Enclose the host group name in double quotation marks (" "). `defaultGroup` is the host group that contains the host to which the virtual disk is mapped.

Notes

A host group is an optional topological element that you can define if you want to designate a collection of hosts that share access to the same virtual disks. The host group is a logical entity. Define a host group only if you have two or more hosts that can share access to the same virtual disks.

You can use any combination of alphanumeric characters, hyphens, and underscores for the names. Names can have a maximum of 30 characters.

The access virtual disk is the virtual disk in a SAN environment that is used for in-band communication between the storage management software and the storage array RAID controller module. This virtual disk uses a LUN address and consumes 20 MB of storage space that is not available for application data storage. An access virtual disk is required only for in-band managed storage arrays. If you specify the `accessVirtualDisk` parameter, the only property you can set is the `logicalUnitNumber` parameter.

Show Blocked Events

Description

This command returns a list of events that are currently blocked by the set `blockEventAlert` command. The events in this list are the events that will not send any notifications that you have configured using the alert terminals and parameters; that is, the email, syslog, and trap notifications.

The output has one line for each event that is blocked, listing the type of event in a hexadecimal format followed by a description of the event. Following is an example of the output:

```
Executing Script... 0x280D Enclosure Failure 0x282B Enclosure Redundancy Lost
Script execution complete.
```

Syntax

```
show blockedEventAlertList
```

Parameters

None

Show Consistency Group Snapshot Image

Description

This command shows one snapshot image or several snapshot images that are in one or more snapshot consistency groups.

Syntax

```
show ((CGSnapImage [(CGSnapImageName | CGSnapImageSequenceNumber)]) |
(CGSnapImages [(CGSnapImageNumber1 ... CGSnapImageNumbern |
CGSnapImageSequenceNumber1 ... CGSnapImageSequenceNumbern)]) | allCGSnapImages
[summary]
```

Parameters

Parameter	Description
`cgSnapImage` or `cgSnapImages`	The name of the snapshot image in a consistency group. The name of a snapshot image is comprised of two parts: • The name of the consistency group. • An identifier for the snapshot image in the consistency group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the consistency group.

Parameter	Description
	• NEWEST - Use this option when you want to show the latest snapshot image created in the consistency group. • OLDEST - Use this option when you want to show the earliest snapshot image created in the consistency group. Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]). You can enter more than one snapshot name or sequence number. Enclose all of the snapshot names in one set of double quotation marks (" ") inside square brackets ([ ]). Separate each snapshot image name with a white space.
allCGSnapImages	The setting to return all of the snapshot images from the consistency groups.
summary	The setting to return a concise list of information about all of the snapshot images in the storage array.

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot consistency group.
- The identifier of the snapshot image.

For example, if you want to show snapshot image 12345 in a snapshot consistency group that has the name **snapCGroup1**, you would use this command:

```
show snapImage ["snapCGroup1:12345"];
```

To show the most recent snapshot image in a snapshot consistency group that has the name **snapCGroup1**, you would use this command:

```
show snapImage ["snapCGroup1:newest"];
```

To show the snapshot images in several snapshot consistency groups that has the names **snapCGroup1**, **snapCGroup2**, and **snapCGroup3**, you would use this command:

```
show snapImages ["snapCGroup1:12345 snapCGroup2:newest snapCGroup3:oldest"];
```

Note that in these examples the snapshot consistency group name is separated from the snapshot image identifier by a colon (:).

Show Current iSCSI Sessions

Description

This command displays information about an iSCSI session for either an iSCSI initiator or iSCSI target.

Syntax

When running from the command line, use:

```
show iscsiInitiator [\"initiatorName\"] iscsiSessions

show iscsiTarget [\"targetName\"] iscsiSessions
```

When running from a script file, use:

```
show iscsiInitiator ["initiatorName"] iscsiSessions

show iscsiTarget ["targetName"] iscsiSessions
```

Parameters

Parameter	Description
`initiatorName`	The name of the iSCSI initiator for which session information is desired. Enclose the iSCSI initiator name in double quotation marks (" "). The name and quotation marks must also be enclosed in either square brackets ([ ]) or angle brackets (< >). If the name is longer than 30 characters, the angle brackets must be used.
`targetName`	The name of the iSCSI target for which session information is desired. Enclose the iSCSI target name in double quotation marks (" "). The name and quotation marks must also be enclosed in either square brackets ([ ]) or angle brackets (< >). If the name is longer than 30 characters, the angle brackets must be used.

NOTE: This command can be entered without defining any arguments, i.e. `show iscsiSessions`. When the command is entered without arguments, the information displayed is for all active iSCSI sessions. To limit the information displayed, enter a specific iSCSI initiator or a specific iSCSI target. The command then displays only information about that particular iSCSI initiator or iSCSI target session.

Show Disk Group

Description

This command returns the following information about a disk group:

- Status (online or offline)
- Physical disk type (SAS)
- Enclosure loss protection (yes or no)
- Current owner (RAID controller module 0 or RAID controller module 1)
- Associated virtual disks and free capacity
- Associated physical disks

NOTE: You can use the free capacity area value when you create a virtual disk based on the free capacity of a disk group. For a description of how to use the free capacity value, see the `create virtualDisk` command on Create RAID Virtual Disk (Free Capacity Base Select).

Syntax

```
show diskGroup [diskGroupNumber]
```

Parameter

Parameter	Description
`diskGroup`	Number of the disk group for which to display information. You must put brackets ([ ]) around the disk group number.

Show Disk Group Export Dependencies

Description

This command shows a list of dependencies for the physical disks in a disk group that you want to move from one storage array to a second storage array.

Syntax

```
show diskGroup [diskGroupName] exportDependencies
```

Parameter

Parameter	Description
`diskGroup`	Number of the disk group for which to display information. You must put brackets ([ ]) around the disk group number.

Show Disk Group Import Dependencies

Description

This command shows a list of dependencies for the physical disks in a disk group that you want to move from one storage array to a second storage array.

Syntax

```
show diskGroup [diskGroupName] importDependencies
[cancelImport=(TRUE | FALSE)]
```

Parameters

Parameter	Description
`diskGroup`	Number of the disk group for which to display information. You must put brackets ([ ]) around the disk group number.
`cancelImport`	The setting to spin the physical disks back down after the disk group dependencies have been read. To spin down the physical disks, set this parameter to `TRUE`. To let the physical disks stay spinning, set this parameter to `FALSE`.

NOTE: This command returns the dependencies of a specific disk group, which must be in an Exported state or a Forced state. If a decision is made to retain the listed dependencies, then the `cancelImport` parameter can be enforced to spin the physical disks back down. You must run the `show diskGroup importDependencies` command before you run the `start diskGroup import` command.

Show Disk Pool

Description

This command returns information on the used and free capacity and other properties for the disk pool.

Syntax

`show diskPool [diskPoolName]`

Parameter

Parameter	Description
`diskPool`	The alphanumberic identifier of the disk pool (including – and _) that you want to set to a Complete state. Enclose the *diskPoolName* in square brackets ([ ]).

Show Host Ports

Description

For all HBA host ports connected to a storage array, this command returns the following information:

- HBA host port identifier
- HBA host port name
- HBA host type

Syntax

```
show allHostPorts
```

Parameters

None.

Additional Information

This command returns HBA host port information similar to this example.

HOST PORT IDENTIFIER	HOST PORT NAME	HOST TYPE
12:34:56:54:33:22:22:22	Jupiter1	Solaris
12:34:56:78:98:98:88:88	Pluto 1	Windows 2000/Server 2003 Clustered
54:32:12:34:34:55:65:66	Undefined	Undefined

Show Physical Disk

Description

For each physical disk in the storage array, this command returns the following information:

- The total number of physical disks
- The type of physical disk (SAS)
- Basic physical disk information:
  - Enclosure location and slot location
  - Status
  - Capacity
  - Data transfer rate
  - Product ID
  - Firmware level
- Physical disk channel information:
  - Enclosure location and slot location
  - Preferred channel
  - Redundant channel
- Hot spare coverage
- Details for each physical disk

Depending on the size of the storage array, this information can be several pages long. The physical disk information is also returned when you issue the `show storageArray profile` command.

Syntax

```
show (allPhysicalDisks
[driveMediaType=(HDD | SSD | unknown | allMedia)] |[physicalDiskType=(SAS)])
```

```
| (physicalDisk [enclosureID,slotID] | physicalDisks [enclosureID1,slotID1
enclosureID2,slotID2... enclosureIDn,slotIDn])
summary
```

Parameters

Parameter	Description
`allPhysicalDisks`	Returns information about all physical disks in the storage array. **NOTE:** To determine information about the type and location of all physical disks in the storage array, use the `allPhysicalDisks` parameter.
`driveMediaType`	retrieve information. Valid physical disk media are these: • HDD—Use this option when you have hard physical disks • in the expansion enclosure. • SSD—Use this option when you have solid state physical disks in the expansion enclosure. • unknown—Use if you are not sure what types of physical disk media are in the expansion enclosure. • allMedia – Use this option when you want to use all types of physical disk media that are in the expansion enclosure.
`physicalDiskType`	The type of physical disk for which you want to retrieve information. The valid physical disk type is SAS.
`physicalDisk or physicalDisks`	Identifies the enclosure and slot where the physical disk resides. You can enter enclosure IDs and slot IDs for one or several physical disks. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put parentheses around the enclosure ID values and the slot ID values.
`summary`	Returns the status, capacity, data rate, product ID, and firmware version for the specified physical disks. **NOTE:** To determine the type of physical disk in a specific location, use the `physicalDisk` parameter, and enter the enclosure and slot IDs for the physical disk.

Show Physical Disk Channel Statistics

Description

This command shows cumulative physical disk channel data transfer and error information. If the RAID controller module has automatically degraded a channel, this command also shows interval statistics. When using this command, you can display information about one specific physical disk channel, several physical disk channels, or all physical disk channels.

Syntax

```
show (physicalDiskChannel [(1 | 2)] |
physicalDiskChannels [1 2] | allPhysicalDiskChannels) stats
```

Parameters

Parameter	Description
`physicalDiskChannel`	Identifier number of the physical disk port for which to display information. Valid physical disk port values are `1` or `2`. You must put brackets ([ ]) around the physical disk channel values.
`physicalDiskChannels`	Identifier number of the physical disk ports for which to display information. Valid physical disk port values are `1` or `2`. You must put brackets ([ ]) around the physical disk channel values.
`allPhysicalDiskChannels`	The identifier that selects all of the physical disk ports.

Show Physical Disk Download Progress

Description

This command returns the status of firmware downloads for the physical disks targeted by the `download physicalDisk firmware` or `download storageArray physicalDiskFirmware` commands.

Syntax

```
show allPhysicalDisks downloadProgress
```

Parameters

None.

NOTE: When all firmware downloads have successfully completed, this command returns a `Successful` status. If any firmware downloads fail, this command shows the firmware download status of each targeted physical disk. This command returns the status values shown in the following table.

Status	Definition
Successful	Downloads completed without errors.
Not Attempted	Downloads did not start.
Partial Download	Downloads are in progress.
Failed	Downloads completed with errors.

Show RAID Controller Module

Description

For each RAID controller module in a storage array, this command returns the following information:

- Status (Online, Offline)
- Current firmware and NVSRAM configuration
- Pending firmware and NVSRAM configuration configurations (if any)
- Board ID
- Product ID
- Product revision
- Serial number
- Date of manufacture
- Date and time to which the RAID controller module is set
- Associated virtual disks (including preferred owner)
- Ethernet port
- Physical disk interface

Syntax

```
show (allControllers | controller [(0 | 1)]) [summary]
```

Parameters

Parameter	Description
`allControllers`	Returns information about both RAID controller modules in the storage array.
`controller`	Returns information about a specific RAID controller module in the storage array. Valid RAID controller module identifier values are 0 and 1. The identifier value is 0 for the RAID controller module on the top and 1 for the RAID controller module on the bottom when viewed from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier.
`summary`	Returns a concise list of information about both RAID controller modules in the storage array.

Show RAID Controller Module NVSRAM

Description

This command returns a list of the NVSRAM byte values for the specified host type. If you do not enter the optional parameters, this command returns a list of all NVSRAM byte values.

Syntax

```
show (allControllers | controller [(0 | 1)]) NVSRAM [hostType=(
hostTypeIndexLabel | host="hostName")]
```

Parameters

Parameter	Description
`allControllers`	Returns information about both RAID controller modules in the storage array.
`controller`	Returns information about a specific RAID controller module in the storage array. Valid RAID controller module identifier values are `0` and `1`. The identifier value is `0` for the RAID controller module on the top and `1` for the RAID controller module on the bottom when viewed from the rear of the enclosure. You must put brackets ([ ]) around the RAID controller module identifier.
`hostType`	Specifies the index label or number of the host type. Use the `show storageArray hostTypeTable` command to generate a list of available host type identifiers.
`host`	Specifies the name of the host connected to the RAID controller modules. You must put quotation marks (" ") around the host name.

NOTE: Use the `show controller NVSRAM` command to display parts or all of the NVSRAM before using the `set controller` command to change NVSRAM values.

Show Remote Replication Group

Description

This command displays configuration information for one or more remote replication groups. This command also displays the remote replicated pairs associated with each remote replication group, including incomplete remote replicated pairs.

You can also use this command to show the progress of periodic data synchronization on all of the replicated pairs within the remote replication group.

Syntax

```
show (allAsyncRemoteReplicationGroups |
asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"])
[summary]
```

Parameters

Parameter	Description
`allAsyncRemoteReplicationGroups`	Use this parameter if you want to display the properties for all remote replication groups.
`asyncRemoteReplicationGroup`	Use this parameter to display the properties for a remote replication group. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets. If the remote replication group name consists only of numbers, such as "1002," you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets.
`summary`	Use this optional parameter to show a concise list of information about the synchronization progress of one or more remote replication groups.

Show Remote Replication Group Synchronization Progress

Description

This command displays the progress of periodic synchronization of the remote replication group between the local and remote storage array. This command returns the progress of data synchronization on all of the replicated pairs within the remote replication group. This command shows the progress as a percentage of data synchronization that has been completed.

NOTE: There are two types of synchronization: initial synchronization and periodic synchronization. Initial remote replication group synchronization progress is displayed in the Long Running Operations dialog and by executing the `show storageArray longRunningOperations` command.

Syntax

```
show  asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
[synchronizationProgress]
```

Parameters

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group for which you want to check the synchronization progress. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets. If the remote replication group name consists only of numbers, such as "1002," you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets.
`synchronizationProgress`	Use this optional parameter to display the periodic synchronization progress of the remote replication group.

Show Remote Replication Virtual Disk Candidates

Description

NOTE: This command valid only with Fibre Channel storage arrays.

This command returns information about the candidate virtual disks on a remote storage array that can be used as secondary virtual disks in a Remote Replication configuration.

Syntax

```
show remoteReplication candidates primary= ”virtualDiskName”
remoteStorageArrayName=storageArrayName
```

Parameters

Parameter	Description
`primary`	The name of the local virtual disk that you want for the primary virtual disk in the remote-replicated pair. Enclose the primary virtual disk name in double quotation marks (“ ”).
`remoteStorageArrayName`	The remote storage array that contains possible virtual disks for a secondary virtual disk. If the remote storage array name has special characters, it must be enclosed in double quotation marks (“ ”).

Show Remote Replication Virtual Disk Synchronization Progress

NOTE: This command valid only with Fibre Channel storage arrays.

Description

This command returns the progress of data synchronization between the primary virtual disk and the secondary virtual disk in a Remote Replication configuration. This command shows the progress as a percentage of data synchronization that has been completed.

Syntax

```
show remoteReplication (localVirtualDisk
[“virtualDiskName”] |
localVirtualDisks [“virtualDiskName1” ...
“virtualDiskNameN”])
```

Parameters

Parameter	Description
`localVirtualDisk or localVirtualDisks`	The name of the primary virtual disk of the remote replication pair for which you want to check synchronization progress. Enclose the primary virtual disk name in double quotation marks (“ ”) inside square brackets ([ ]).

Show SNMP Communities

Description

This command returns information about the Simple Network Management Protocol (SNMP) communities defined for the storage array. SNMP communities are collections of devices, such as routers, switches, printers, and storage devices grouped together to manage and monitor the devices. this command displays the following information about the SNMP communities.

- The total number of communities
- The total number of trap destinations
- The names of the SNMP communities

Syntax

```
show (allSnmpCommunities |
snmpCommunity communityName="snmpCommunityName" |
snmpCommunities ["snmpCommunityName1" ... "snmpCommunityNameN"])
[summary]
```

Parameters

Parameter	Description
`allSnmpCommunities`	This parameter returns information about all of the SNMP communities in the storage array.
`communityName`	The name of the SNMP community for which you want to retrieve information. Enclose the SNMP community name in double quotation marks (" ").
`snmpCommunities`	The names of several SNMP communities for which you want to retrieve information. Enter the names of the SNMP communities using these rules: • Enclose all of the names in square brackets ([ ]). • Enclose each of the names in double quotation marks (" "). • Separate each of the names with a space.
`summary`	This parameter returns a list of the total number of SNMP communities and the total number of SNMP trap destinations. When you use this parameter information detail is omitted.

Notes

SNMP supports one or more communities to which managers and managed devices belong. SNMP requests contain a community string in the data packets passed over the network that acts as a crude password. SNMP agents can reject GET or SET requests with an unrecognized community string. A community string is also included in TRAP notifications sent from agent to manager as well.

The embedded SNMP agent supports both IPV4 and IPV6.

This command returns SNMP community information similar to this example.

```
SNMP COMMUNITIES----------------------------- SUMMARY
Total SNMP Communities: 2
Total SNMP Trap Destinations: 1
DETAILS
SNMP Community: TestComm1
SNMP Permission: Read Only
        Associated Trap Destination:
        Trap Receiver IP Address: 10.113.173.133
        Send Authentication Failure Traps: true
SNMP Community: Test2
SNMP Permission: Read Only Associated Trap Destination:
```

Minimum Firmware Level

8.10

Show SNMP MIB II System Group Variables

Description

This command returns information about the Simple Network Management Protocol (SNMP) system variables. The system variables are maintained in a Management Information Base II (MIB-II) database.

Syntax

```
show snmpSystemVariables
```

Parameters

None.

Notes

This command returns SNMP system variables information similar to this example.

```
SYSTEM VARIABLES
Sys Name: NONE
Sys Contact: NONE Sys
Location: NONE
```

Minimum Firmware Level

8.10

Show Snapshot Group

Description

This command returns this information about one or more snapshot image groups.

Syntax

```
show (allSnapGroups | snapGroup [snapGroupName] |
snapGroups [snapGroupName1 ... snapGroupNamen])
[summary | schedule]
```

Parameters

Parameter	Description
`allSnapGroups`	The parameter to return information about all of the snapshot groups in the storage array.
`snapGroup` or `snapGroups`	The name of the specific snapshot group for which you are retrieving information. Enclose the snapshot group name in square brackets ([ ]).

Parameter	Description
	If you enter more than one snapshot group name, separate each name with a space. Enclose all of the snapshot group names in only one set of double quotation marks (" ").
summary	The parameter to return a concise list of information about the snapshot groups.
schedule	The parameter to return a concise list of information about the schedules for the snapshot group copy operations.

Notes

This command returns snapshot group information as shown in this example:

```
SNAPSHOT GROUP DETAILS
SNAPSHOT GROUPS------------------------------
SUMMARY
Total Snapshot Groups: 1
Total Snapshot Images: 0
Enabled Snapshot Image Schedules: 0
Status: 1 Optimal, 0 Non Optimal

Name                Type                Status              Associated Base
                                                            Virtual Disk

2_SG_01             Standard            Optimal             2

Total               Available           Snapshot Image      Snapshot        Scheduled
Repository          Repository          Limit               Images
Capacity            Capacity

10.318 GB           10.318 GB           0                   0               No
                    (100%)

DETAILS

Snapshot Group "2_SG_01"
Status: Optimal
Type: Standard
Associated base virtual disk: 2
Cluster size: 65,536 bytes

Repository
Total repository virtual disks: 3
Aggregate repository status: Optimal
Total repository capacity: 10.318 GB
Used capacity: 0.000 MB (0%)
Available repository capacity: 10.318 GB (100%)
Repository full policy: Auto-purge Snapshot Images
Utilization alert threshold: 75%

Snapshot images
Total Snapshot images: 0
```

```
Auto-delete Snapshot images: Disabled
Snapshot image schedule: Not Applicable
```

Show Snapshot Image

Description

This command returns information about the snapshot images that a user had previously created.

Syntax For Showing A Specific Snapshot Image

```
show (snapImage ["snapImageName"] |
snapImages ["snapImageName1 ... snapImageNamen"] |
allSnapImages])
[summary]
```

Parameters

Parameter	Description
`snapImage` or `snapImages`	The name of the snapshot image. The name of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]). You can enter more than one snapshot name or sequence number. Enclose all of the snapshot names in one set of double quotation marks (" ") inside square brackets ([ ]). Separate each snapshot image name with a white space.
`allSnapImages`	The parameter to return information about all of the snapshot images in the storage array.
`summary`	This parameter returns a concise list of information about the snapshot images.

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group.
- The identifier of the snapshot image.

For example, if you want to show snapshot image 12345 in a snapshot group that has the name **snapGroup1**, you would use this command:

```
show snapImage ["snapGroup1:12345"];
```

To show the most recent snapshot image in a snapshot group that has the name **snapGroup1**, you would use this command:

```
show snapImage ["snapGroup1:newest"];
```

To show the snapshot images in several snapshot consistency groups that has the names **snapGroup1**, **snapGroup2**, and **snapGroup3**, you would use this command:

```
show snapImages ["snapGroup1:12345 snapGroup2:newest snapGroup3:oldest"];
```

Show Snapshot Virtual Disks

Description

This command returns information about one or more snapshot virtual disks.

Syntax

```
show (allSnapVirtualDisks | snapVirtualDisk [snapVirtualDiskName] |
snapVirtualDisks [snapVirtualDiskName1 ... snapVirtualDiskNamen])
[summary]
```

Parameters

Parameter	Description
`allSnapVirtualDisks`	The parameter to return information about all of the snapshot virtual disks in the storage array.
`snapVirtualDisk` or `snapVirtualDisks`	The alphanumeric identifier (including - and _) of a specific snapshot virtual disk about which you are retrieving information. Enclose the snapshot virtual disk name in square brackets ([ ]). When you enter more than one snapshot virtual disk identifier, separate the identifiers with a space. Enclose all of the identifiers in only one set of square brackets ([ ]).
`summary`	The parameter to return a concise list of information about the snapshot virtual disks.

Notes

This command returns snapshot virtual disk information as shown in this example:

```
SNAPSHOT Virtual Disk(SNAPSHOT-IMAGE BASED) SUMMARY
SUMMARY

Total Snapshot virtual disks: 1
    Most Recent Snapshot virtual disk: Day month date hh:mm:ss yyyy
    Status: 1 Optimal, 0 Non Optimal

Name            Type            Status          Capacity        Associated
                                                                Base Virtual
                                                                Disk

2_SV_0001       Standard        Optimal         3.000 GB        2

Snapshot Virtual Disk      Snapshot Image Timestamp   Mode
Timestamp

1/23/12 6:44:31 PM IST     1/23/12 6:27:36 PM IST     Read Write

Total Repository Capacity              Available Repository Capacity

1.199 GB                               0.125 MB (0%)
```

The size of your monitor determines how the information wraps and will affect how the information appears.

Show SSD Cache

Description

This command displays information about the SSD cache.

Syntax

```
show ssdCache [ssdCacheName]
```

Parameter

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache for which you want to get information. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Notes

This command returns the SSD cache information similar to this example.

```
SSD Cache name: my_cache

Status:                                 Optimal
Type:                                   Read Only
I/O characteristic type:                File System
Maximum capacity allowed:               1,862.645 GB
Current capacity:                       557.792 GB
Additional capacity allowed             1,304.852 GB
Physical disk capacities:               All 278.896 GB

Data Service (DS) Attributes
Security capable:                       No
Secure:                                 No
Data Assurance (DA) capable:            No

Associated physical disks:
Enclosure                               Slot
0                                       4
0                                       11
Virtual disks using SSD cache:          virtualDisk_test
```

Show SSD Cache Statistics

Description

This command displays data about the SSD cache usage.

Syntax

```
show ssdCache [ssdCacheName] ssdCacheStatistics [controller=(0|1|both)]
[file="filename"]
```

Parameters

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache for which you want to get information. Enclose the identifier in square

Parameter	Description
	brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
`controller`	Each RAID controller module stores SSD cache metadata for the virtual disks that it owns. Therefore, the SSD cache statistics are maintained and displayed per RAID controller module. Valid RAID controller module identifiers are `0`, `1`, or both, where `0` is the RAID controller module in slot 0, `1` is the RAID controller module in slot 1, and both is both RAID controller modules. Enclose the RAID controller module identifier in square brackets ([ ]). If you do not specify a RAID controller module, the default value is `both`.
`file`	The file path and the file name to which you want to save the SSD cache statistics. Additional statistics are available when you save the statistics to a file. Enclose the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI\logs \statistics.csv"**.

Notes

The following statistics are displayed on the screen:

Reads Total number of host reads of SSD cache-enabled virtual disks.

Compare the **Reads** relative to **Writes**. The **Reads** need to be greater than the **Writes** for effective SSD cache operation. The greater the ratio of Reads to Writes the better the operation of the cache.

Writes Total number of host writes to SSD cache-enabled virtual disks.

Cache Hits A count of the number of cache hits.

Cache Hits (%) Derived from Cache Hits/total reads.

The Cache Hit percentage should be greater than 50% for effective SSD cache operation. A small number could be indicative of several things:

- Ratio of Reads to Writes is too small
- Reads are not repeated
- Cache capacity is too small



NOTE: To help determine the ideal SSD cache capacity, you can run the Performance Modeling Tool using the `start ssdCache [ssdCacheName] performanceModeling` command.

Cache Allocation (%)	The amount of SSD cache storage that is allocated, expressed as a percentage of the SSD cache storage that is available to this RAID controller module. Derived from allocated bytes / available bytes. Cache Allocation percentage normally shows as 100%. If this number is less than 100%, it means either the cache has not been warmed or the SSD cache capacity is larger than all the data being accessed. In the latter case, a smaller SSD cache capacity could provide the same level of performance. Note that this does not indicate that cached data has been placed into the SSD cache, it is simply a preparation step before data can be placed in the SSD cache.
Cache Utilization (%)	The amount of SSD cache storage that contains data from enabled virtual disks, expressed as a percentage of SSD cache storage that is allocated. This value represents the utilization or density of the SSD cache. Derived from user data bytes / allocated bytes. Cache Utilization percentage normally is lower than 100%, perhaps much lower. This number shows the percent of SSD cache capacity that is filled with cache data. The reason this number is lower than 100% is that each allocation unit of the SSD cache, the SSD cache cache-block, is divided into smaller units called sub-blocks, which are filled somewhat independently. A higher number is generally better, but performance gains can be significant even with a smaller number.

These additional statistics are included when you save the data to a file:

Read Blocks	Number of blocks in host reads.
Write Blocks	Number of blocks in host writes.
Full Hit Blocks	Number of block cache hits. The full hit blocks indicate the number of blocks that have been read entirely from SSD cache. The SSD cache is only beneficial to performance for those operations that are full cache hits.
Partial Hits	Number of host reads where at least one block, but not all blocks, were in the SSD cache. This is an SSD cache miss where the reads were satisfied from the base virtual disk. Partial cache hits and partial cache hit blocks result from an operation that has only a portion of its data in the SSD cache. In this case, the operation must get the data from the cached HDD virtual disk. The SSD cache offers no performance benefit for this type of hit. If the partial cache hit blocks count is higher than the full cache hit blocks, it is possible that a different I/O characteristic type (file system, database, or web server) could improve the performance.
Partial Hits - Blocks	Number of blocks in Partial Hits. Partial cache hits and partial cache hit blocks result from an operation that has only a portion of its data in the SSD cache. In this case, the operation must get the data from the cached HDD virtual disk. The SSD cache offers no performance benefit for this type of hit. If the partial cache hit blocks count is higher than the full cache

	hit blocks, it is possible that a different I/O characteristic type (file system, database, or web server) could improve the performance.
Misses	Number of host reads where none of the blocks were in the SSD cache. This is an SSD cache miss where the reads were satisfied from the base virtual disk.
Misses - Blocks	Number of blocks in Misses.
Populate Actions (Host Reads)	Number of host reads where data was copied from the base virtual.
Populate Actions (Host Reads) - Blocks	Number of blocks in Populate Actions (Host Reads).
Populate Actions (Host Writes)	Number of host writes where data was copied from the base virtual disk to the SSD cache. The **Populate Actions (Host Writes)** count may be zero for the cache configuration settings that do not fill the cache as a result of a Write I/O operation.
Populate Actions (Host Writes) - Blocks	Number of blocks in **Populate Actions (Host Writes)**.
Invalidate Actions	Number of times data was invalidated/removed from the SSD cache. A cache invalidate operation is performed for every host write request, every host read request with Forced Unit Access (FUA), every verify request, and in some other circumstances.
Recycle Actions	Number of times that the SSD cache block has been reused for another base virtual disk and/or a different LBA range. For effective cache operation, it is important that the number of recycles is small compared to the combined number of read and write operations. If the number of Recycle Actions is close to the combined number of Reads and Writes, then the SSD cache is thrashing. Either the cache capacity needs to be increased or the workload is not favorable for use with SSD cache.
Available Bytes	Number of bytes available in the SSD cache for use by this RAID controller module. The available bytes, allocated bytes, and user data bytes are used to compute the Cache Allocation % and the Cache Utilization %.
Allocated Bytes	Number of bytes allocated from the SSD cache by this RAID controller module. Bytes allocated from the SSD cache may be empty or they may contain data from base virtual disks. The available bytes, allocated bytes, and user data bytes are used to compute the Cache Allocation % and the Cache Utilization %.
User Data Bytes	Number of allocated bytes in the SSD cache that contain data from base virtual disks.

The available bytes, allocated bytes, and user data bytes are used to compute the Cache Allocation % and the Cache Utilization %.

Show Storage Array

NOTE: With firmware version 8.10 the cacheFlushStop parameter is deprecated and will be removed in a future release of storage management software.

Description

This command returns configuration information about the storage array. The parameters return lists of values for the components and features in the storage array. You can enter the command with a single parameter or more than one parameter. If you enter the command with no parameters, the entire storage array profile is displayed (which is the same information as if you entered the `profile` parameter).

Syntax

```
show storageArray (autoSupportConfig | profile | batteryAge
| connections | defaultHostType | healthStatus | hostTypeTable |
hotSpareCoverage | features | time | virtualDiskDistribution
longRunningOperations
| summary)
```

Parameters

Parameter	Description
`profile`	Displays all properties of the logical and physical components that comprise the storage array. The information returned takes several screens to display. NOTE: The profile parameter returns detailed information about the storage array. The information covers several screens on a display. You might need to increase the size of your display buffer to see all the information. Because this information is so detailed, you might want to save the output of this parameter to a file. To save the output to a file, enter the `show storageArray` command similar to the following: `smcli 123.45.67.89 -c "show storagearray profile;" -o "c:\folder \storagearray profile.txt"`

Parameter	Description
	NOTE: The previous command syntax is for a host running Windows. The actual syntax varies depending on your operating system.
`autoSupportConfig`	The parameter to return information about the current state of the operation to automatically collect support data. The information this parameter returns is: • Whether the operation is enabled or disabled • The location of the folder where the support data file is located
`batteryAge`	Displays the status, the age of the battery in days, and the number of days until the battery needs to be replaced.
`connections`	Displays a list of the drive channel port locations and the drive channel connections.
`defaultHostType`	Displays the default host type and host type index.
`healthStatus`	Displays the health, logical properties, and physical component properties of the storage array.
`hostTypeTable`	Displays a table of all host types known to the RAID controller module. Each row in the table displays a host type index and the platform the index represents.
`hotSpareCoverage`	Displays information about which virtual disks of the storage array have hot spare coverage and which virtual disks do not.
`features`	Displays a list of the feature identifiers for all enabled features in the storage array.
`time`	Displays the current time to which both RAID controller modules in the storage array are set.
`virtualDiskDistribution`	Displays the current RAID controller module owner for each virtual disk in the storage array.
`summary`	Returns a concise list of information about the storage array configuration.
`longRunningOperations`	Shows the long running operations for each disk group and each virtual disk in the storage array. The `longRunningOperation` parameter returns this information: • Name of the disk group or virtual disk • Long running operation • Status • % complete

Parameter	Description
	• Time remaining

NOTE: When you save the information to a file, you can use the information as a record of your configuration and as an aid during recovery.

Show Storage Array Auto Configure

Description

This command shows the default autoconfiguration that the storage array creates if you issue the `autoConfigure storageArray` command. To determine whether the storage array can support specific properties, enter the parameter for the properties when you issue this command. You do not, however, need to enter any parameters for this command to return configuration information. If you do not specify any properties, this command returns the RAID 5 candidates for each physical disk type. If RAID 5 candidates are not available, this command returns candidates for RAID 1 or RAID 0. In order to view the RAID 6 autoConfiguration options, you must specify the `raidLevel` option. When you specify auto-configuration properties, the RAID controller modules validate that the firmware can support the properties.

Syntax

```
show storageArray autoConfiguration
[physicalDiskType=(SAS)
raidLevel=(0 | 1 | 5 | 6)
diskGroupWidth=numberOfPhysicalDisks diskGroupCount=numberOfDiskGroups
virtualDisksPerGroupCount=
numberOfVirtualDisksPerGroup
hotSpareCount=numberOfHotspares
segmentSize=segmentSizeValue
cacheReadPrefetch=(TRUE | FALSE)
securityType=(none | capable | enabled)]
```

Parameters

Parameter	Description
`physicalDiskType`	Type of physical disk to use for the storage array. Valid physical disk type is SAS. This parameter is not required if only one type of physical disk is in the storage array.
`raidLevel`	RAID level of the disk group that contains the physical disks in the storage array. Valid RAID levels are `0`, `1`, `5` or `6`.
`diskGroupWidth`	Number of physical disks in a disk group in the storage array. This number depends on the capacity of the physical disks. Integer values are required. For information about the number of physical disks that you can use in a disk group, see Enclosure Loss Protection.

Parameter	Description
diskGroupCount	Number of disk groups in the storage array. Use integer values.
virtualDisksPerGroupCount	Number of equal-capacity virtual disks per disk group. Use integer values.
hotSpareCount	Number of hot spares desired in the storage array. Use integer values. For information about hot spares, see Assigning Global Hot Spares.
segmentSize	Amount of data (in kilobytes) that the RAID controller module writes on a single physical disk in a virtual disk before writing data on the next physical disk. Valid values are 8, 16, 32, 64, 128, 256, or 512. For information about the segmentSize parameter, see Using The Auto Configure Command.
cacheReadPrefetch	The setting to turn on or turn off cache read prefetch. To turn off cache read prefetch, set this parameter to FALSE. To turn on cache read prefetch, set this parameter to TRUE.
securityType	The setting to specify the security level when creating the disk groups and all associated virtual disks. These settings are valid: • none—The disk group and virtual disks are not secure. • capable—The disk group and virtual disks are capable of having security set, but security has not been enabled. • enabled—The disk group and virtual disks have security enabled.

Show Storage Array Core Dump

Description

This command show details of the core dump on the RAID controller module cache, if a core dump is available.

Syntax

```
show storageArray coreDump
```

Parameters

None.

Show Storage Array DBM Database

Description

This command retrieves and shows metadata for the on-board backup locations of a storage array. When there are multiple backup locations, metadata is shown for each location.

Syntax

```
show storageArray dbmDatabase
```

Parameters

None.

Show Storage Array Host Topology

Description

This command returns the storage partition topology, the host type labels, and the host type index for the host storage array.

Syntax

```
show storageArray hostTopology
```

Parameters

None.

NOTE:

This command returns the host topology information similar to this example.

TOPOLOGY DEFINITIONS

DEFAULT GROUP

Default type: Windows 2000/Server 2003 Non-Clustered

Host Group: scott

Host: scott1

Host Port: 28:37:48:55:55:55:55:55

Alias: scott11

Type: Windows 2000/Server 2003 Clustered

Host: scott2

Host Port: 98:77:66:55:44:33:21:23

Alias: scott21

Type: Windows 2000/Server 2003 Clustered

Host: Bill

Host Port: 12:34:55:67:89:88:88:88

Alias: Bill1

Type: Windows 2000/Server 2003 Non-Clustered

NVSRAM HOST TYPE INDEX DEFINITIONS

HOST TYPE ALUA/AVT STATUS ASSOCIATED INDEXS

Linux (DM-MP) Disabled 6

VMWare Enabled 10 (Default)

Windows Enabled 1

Show Storage Array LUN Mappings

Description

This command returns information from the storage array profile about the storage array LUN mappings. If you run this command with no parameters, this command returns all LUN mappings.

Syntax

```
show storageArray lunMappings (host ["hostName"] | hostgroup ["hostGroupName"])
```

Parameters

Parameter	Description
`host`	Name of a specific host for which to see the LUN mappings. You must put quotation marks (" ") and brackets ([ ]) around the host name. The host name and quotation marks must be inside the brackets.
`hostGroup`	Name of a specific host group for which to see the LUN mappings. You must put quotation marks (" ") and brackets ([ ]) around the host group name. The host group name and quotation marks must be inside the brackets.

Show Storage Array Negotiation Defaults

Description

This statement returns information about connection-level settings that are subject to initiator-target negotiation.

Syntax

```
show storageArray iscsiNegotiationDefaults
```

Parameters

None.

NOTE: Information returned includes RAID controller module default settings (settings that are the starting point for negotiation), and the current active settings.

Show Storage Array ODX Setting

This command returns the current settings for Offloaded Data Transfer (ODX) and VMware vStorage API Array Architecture (VAAI) on the storage array. The settings can be one of these:

True – ODX and VAAI are turned on.

False – ODX and VAAI are turned off.

Inconsistent – The RAID controller modules do not have the same settings.

Unknown – The setting for ODX or VAAI cannot be determined.

Add your section content here.

Syntax

```
show storageArray odxsetting
```

Parameters

None.

NOTE: This command returns the ODX and VAAI information similar to this example.
Windows ODX Setting Status

odxEnabled True | False | Inconsistent | Unknown

vaaiEnabled True | False | Inconsistent | Unknown

Show Storage Array Power Information

Description

This command returns information about the amount of power consumed by the entire storage array and each enclosure in the storage array.

Syntax

```
show storageArray powerInfo
```

Parameters

None

Notes

This command returns power consumption information similar to this example.

```
total power drawn: 310 watts
number of enclosures: 1
enclosure power input details:
enclosure id     power supply serial number     input power
0                         0                                  160 watts
0                         1                                  150 watts
```

Show Storage Array Unconfigured iSCSI Initiators

Description

This command returns a list of initiators that have been detected by the storage array but are not yet configured into the storage array topology.

Syntax

```
show storageArray unconfiguredIscsiInitiators
```

Parameters

None.

Show Storage Array Unreadable Sectors

Description

This command returns a table of the addresses of all of the sectors in the storage array that cannot be read. The table is organized with column headings for the following information:

- Virtual disk user label
- Logical unit number (LUN)
- Accessible by (host or host group)
- Date/time
- Virtual disk-relative logical block address (hexadecimal format – 0xnnnnnnnn)
- Physical disk location (enclosure t, slot s)
- Physical disk-relative logical block address (hexadecimal format – 0xnnnnnnnn)
- Failure type

The data is sorted first by the virtual disk user label and second by the logical block address (LBA). Each entry in the table corresponds to a single sector.

Syntax

```
show storageArray unreadableSectors
```

Parameters

None.

Show String

Description

This command shows a string of text from a script file. This command is similar to the echo command in MS-DOS and UNIX.

Syntax

```
show "textString"
```

Parameters

None.

NOTE: You must enclose the string in double quotation marks (" ").

Show Thin Virtual Disk

Description

This command returns the expansion history or the consumed capacity for the specified thin virtual disk or virtual disks.

Syntax

```
show (allVirtualDisks | virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ... virtualDiskNameN])
(consumedCapacity | (expansionHistory [file=fileName]))
```

Parameters

Parameter	Description
`allVirtualDisks`	The setting to return information about all of the virtual disks in the storage array.
`virtualDisk` or `virtualDisks`	The name of the specific virtual disk for which you are retrieving information. You can enter more than one virtual disk name. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").
`consumedCapacity`	The setting to return a concise list of information about the consumed capacity of the thin virtual disks.
`expansionHistory`	The setting to return a concise list of information about the expansion history of the thin virtual disks.
`file`	The `file` parameter specifies a file to log the output of the `expansionHistory` parameter. The file is valid only when used with the `expansionHistory` parameter. An invalid file name causes the command to fail.

Notes

With the `expansionHistory` parameter, the command returns information similar to the example shown below.

```
Thin virtual disk name: virtual disk-nameRepository virtual disk Name:
REPOS_NNNN

Logged Time          Expansion Type      Start Capacity      End Capacity
```

```
MM/DD/YYYY          Manual|Automatic    NNNNNNNN bytes      NNNNNNNN bytes
HH:MM:SS
```

With the `consumedCapacity` parameter, the command returns information similar to the example shown below.

```
Virtual Disk     Physical         Consumned        Quota           % Prov.
                 Capacity         Capacity                         Consumed

virtualDiskNam   500.000 GB       230.000 GB       700.000 GB      46%
e
```

Show Virtual Disk

Description

For the virtual disks in a storage array, this command returns the following information:

- The number of virtual disks
- The name
- The status
- The capacity
- The RAID level
- The disk group where the virtual disk is located
- Details:
  - The virtual disk ID
  - The subsystem ID
  - The physical disk type (SAS)
  - Enclosure loss protection
  - The preferred owner
  - The current owner
  - The segment size
  - The modification priority
  - The read cache status (enabled or disabled)
  - The write cache status (enabled or disabled)
  - The write cache without batteries status (enabled or disabled)
  - The write cache with replication status (enabled or disabled)
  - The flush write cache after time
  - The cache read prefetch setting (TRUE or FALSE)
  - The enable background media scan status (enabled or disabled)
  - The media scan with redundancy check status (enabled or disabled)
- The snapshot (legacy) repository virtual disks
- The replication repository virtual disks
- The snapshot (legacy) virtual disks
- The snapshot (legacy) copies

Syntax

```
show (allVirtualDisks | virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1 ...virtualDiskNameN]) summary
```

Parameters

Parameter	Description
`allvirtualDisks`	The setting to return information about all of the virtual disks in the storage array.
`virtualDisk or virtualDisks`	The name of the specific virtual disk for which you are retrieving information. You can enter more than one virtual disk name. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, you also must enclose the virtual disk name in double quotation marks (" ").
`summary`	The setting to return a concise list of information about the virtual disks.

Notes

For snapshot virtual disk copies, the `show virtualDisk` command returns information about the schedules for the snapshot virtual disk copies. The schedule information is in this form:

```
Schedule State: "Active" | "Disabled" | "Completed"
Last Run Time: <mm/dd/yyyy> <hh:mm a.m. | p.m.>
Next Run Time: <mm/dd/yyyy> <hh:mm a.m. | p.m.>
Start Date: <mm/dd/yyyy> End Date: <mm/dd/yyyy> | "No End Date"
End Date:
Days of Week: <Sunday - Saturday>, <Sunday - Saturday>, ….
Times for snapshot recreate: <hh:mm a.m. | p.m.>, <hh:mm a.m. | p.m.>
```

Show Virtual Disk Action Progress

Description

For a long-running operation that is currently running on a virtual disk, this command returns information about the virtual disk action and amount of the long-running operation completed. The completed amount of the long-running operation is shown as a percentage (for example, 25 means that 25 percent of the long-running operation is completed).

Syntax

```
show virtualDisk ["virtualDiskName"] actionProgress
```

Parameters

Parameter	Description
`virtualDisk`	Name of the virtual disk running the long-running operation. You must put quotation marks (" ") and brackets ([ ]) around the virtual disk name. The virtual disk name and quotation marks must be inside the brackets.

Show Virtual Disk Copy

Description

This command returns information about virtual disk copy operations. The following information is returned:

- Copy status
- Start timestamp
- Completion timestamp
- Copy priority
- Source virtual disk WWID or target virtual disk WWID
- Read-only attribute setting of the target virtual disk

You can retrieve information about a specific virtual disk copy pair or all virtual disk copy pairs in the storage array.

Syntax

```
show virtualDiskCopy (allVirtualDisks |
source ["sourceName"] |
target ["targetName"])
```

Parameters

Parameter	Description
`allVirtualDisks`	Returns information about virtual disk copy operations for all virtual disk copy pairs.
`source`	Name of the source virtual disk about which to retrieve information. You must put quotation marks (" ") and brackets ([ ]) around the source virtual disk name. The source virtual disk name and quotation marks must be inside the brackets.
`target`	Name of the target virtual disk about which to retrieve information. You must put quotation marks (" ") and brackets ([ ]) around the target virtual disk name. The target virtual disk name and quotation marks must be inside the brackets.

Show Virtual Disk Copy Source Candidates

Description

This command returns information about the candidate virtual disks that you can use as the source for a virtual disk copy operation.

Syntax

```
show virtualDiskCopy sourceCandidates
```

Parameters

None.

Additional Information

This command returns virtual disk copy source information as shown in this example.

```
Virtual Disk Name: finance
Capacity: 4.0 GB
Disk Group: 1
Virtual Disk Name: engineering
Capacity: 4.0 GB
Disk Group: 2
```

Show Virtual Disk Copy Target Candidates

Description

This command returns information about the candidate virtual disks that you can use as the target for a virtual disk copy operation.

Syntax

```
show virtualDiskCopy source ["sourceName"] targetCandidates
```

Parameters

Parameter	Description
`source`	Name of the source virtual disk for which you are trying to find a candidate target virtual disk. You must put quotation marks (" ") and brackets ([ ]) around the source virtual disk name. The source virtual disk name and quotation marks must be inside the brackets.

Show Virtual Disk Performance Statistics

Description

This command returns information about the performance of the virtual disks in a storage array.

Syntax

```
show (allVirtualDisks | virtualDisk [virtualDiskName]
virtualDisks [virtualDiskName1... virtualDiskNameN]) performanceStats
```

Parameters

Parameter	Description
`allvirtualDisks`	The setting to return performance statistics about all of the virtual disks in the storage array.
`virtualDisk or virtualDisks`	Name of the source virtual disk for which you are trying to find a candidate target virtual disk. You must put quotation marks (" ") and brackets ([ ]) around the source virtual disk name. The source virtual disk name and quotation marks must be inside the brackets.

Show Virtual Disk Reservations

Description

This command returns information about the virtual disks that have persistent reservations.

Syntax

```
show (allVirtualDisks | virtualDisk [virtualDiskName]
|
virtualDisks [virtualDiskName1... virtualDiskNameN] reservations
```

Parameters

Parameter	Description
`allVirtualDisks`	Returns reservation information about all virtual disks in the storage array.
`virtualDisk or virtualDisks`	Name of the specific virtual disk for which you are retrieving reservation information. You can enter more than one virtual disk name. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has special characters, you must also put quotation marks (" ") around the virtual disk name.

Start Configuration Database Diagnostic

Description

This command starts a diagnostic test to validate the configuration database in the RAID controller module firmware.

Syntax

```
start storageArray configDbDiagnostic
```

Parameters

None.

Notes

Upon completion of the diagnostic test, the RAID controller module firmware returns one of these results:

- `Diagnosis completed without errors. No ZIP file created.`
- `Diagnosis completed with errors. Refer to the ZIP file created at:` **...\Install_dir\data\FirmwareUpgradeReports\timestamp_buildNo.zip**

If the diagnostic test detects an inconsistency in the configuration database, the RAID controller module firmware performs these actions:

- Returns a description of the inconsistency.
- Saves a ZIP file containing raw binary data.
  The RAID controller module firmware saves the ZIP file to this location:

  ...\Install_dir\data\FirmwareUpgradeReports\timestamp_buildNo.zip

The binary data can be used to help determine the cause of the problem, or the file containing the binary data can be sent to a Customer and Technical Support representative.

To stop the database configuration diagnostic test, use the `stop storageArray configDbDiagnostic` command.

In addition, the database configuration diagnostic test can be started using the storage management software GUI but cannot be stopped using the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

Start Consistency Group Snapshot Rollback

Description

This command starts a rollback operation to the member base virtual disks in a snapshot consistency group. The content of the base virtual disks changes immediately to match the point-in-time content of the consistency group snapshot virtual disk. The base virtual disks immediately becomes available for

read/write requests after the rollback operation has successfully completed. To stop a snapshot consistency group rollback operation, use the `stop rollback` command for consistency groups.

The repository virtual disk that is associated with the consistency group snapshot virtual disk continues to track any new changes between the base virtual disk and the consistency group snapshot virtual disk that occur after the rollback operation is completed.

To stop a rollback operation to the member base virtual disks use the `stop cgSnapImage rollback` command.

Syntax

```
start cgSnapImage ["consistencyGroupName"] rollback
[memberVirtualDiskSet ["memberVirtualDiskName1 ... memberVirtualDiskNamen"]]
```

Parameters

Parameter	Description
`consistencyGroupName`	The name of the consistency group for which you want to start a rollback operation. Enclose the name of the consistency group in double quotation marks (" ") inside square brackets ([ ]). When the `memberVirtualDiskSet` parameter is not used, the rollback process applies to all member virtual disks of the consistency group.
`memberVirtualDiskSet`	The name of one or more member base virtual disks in a consistency group that you want to rollback. Enclose each member base virtual disk name in double quotation marks (" ") inside parentheses(( )). You can enter more than one member virtual disk name. Enclose all of the member virtual disk names in one set of double quotation marks (" ") inside parentheses(( )). Separate each member virtual disk name with a white space.

Notes

To start a roll back operation for an entire consistency group that has the name congroup1, you would use this command:

```
start cgSnapImage ["congroup1"] rollback;
```

To start a roll back operation for base virtual disk members memVol1, memVol2, and memVol3 in a consistency group that has the name **congroup2**, you would use this command:

```
start cgSnapImage ["CG2:12345"] rollback memberVirtualDiskset=("memVol1 memVol2
memVol3");
```

Start Disk Group Blink

Description

This command identifies the physical disks that are logically grouped together to form the specified disk group by blinking the indicator LEDs on the physical disks. (Use the `stop diskGroup blink` command to turn off the indicator LEDs on the physical disks.)

Syntax

```
start diskGroup [diskGroupNumber] blink
```

Parameters

Parameter	Description
`diskGroup`	Number of the disk group to identify. You must put brackets ([ ]) around the disk group number.

Start Disk Group Defragment

Description

This command starts a defragment operation on the specified disk group.

NOTE: Defragmenting a disk group starts a long-running operation that you cannot stop.

Syntax

```
start diskGroup [diskGroupNumber] defragment
```

Parameters

Parameter	Description
`diskGroup`	Number of the disk group to defragment. You must put brackets ([ ]) around the disk group number.

Start Disk Group Export

Description

This command moves a disk group into an **Exported** state. Then you can remove the physical disks that comprise the disk group and reinstall the physical disks in a different storage array.

NOTE: Within the disk group, you cannot move virtual disks that are associated with the premium features from one storage array to another storage array.

Syntax

`start diskGroup [diskGroupName] export`

Parameters

Parameter	Description
`diskGroup`	Number of the disk group to defragment. You must put brackets ([ ]) around the disk group number.

Additional Information

When this command is successful, you can run the `start diskGroup import` command to finish moving the disk group to a Complete state, which makes the disk group available to the new storage array.

If this command is unsuccessful because hardware problems prevented the completion of the export, use the `set diskGroup forceState` command. The `set diskGroup forceState` command lets you use the `start diskGroup import` command to import a disk group.

After the disk group is in an Exported state or a Forced state, you can remove the physical disks that comprise the disk group from the storage array. You can reinstall the physical disks in a different storage array.

Start Disk Group Import

Description

This command moves a disk group into a Complete state to make a newly introduced disk group available to its new storage array. The disk group must be in an Exported state or a Forced state before you run this command. Upon successfully running the command, the disk group is operational.

NOTE: Within the disk group, you cannot move virtual disks that are associated with the premium features from one storage array to another storage array.

Syntax

`start diskGroup [diskGroupName] import`

Parameters

Parameter	Description
`diskGroup`	Number of the disk group to defragment. You must put brackets ([ ]) around the disk group number.

NOTE: Higher-level virtual disks that are specifically related to premium features (Snapshot, Remote Virtual Disk Replication, Virtual Disk Copy, mapping, and persistent reservations) are removed as part of the import operation. You must run the show `diskGroup importDependencies` command before you run the `start diskGroup import` command.

Start Disk Pool Locate

Description

This command initiates a location operation on all of the physical disks that comprise the specified disk pool.

Syntax

```
start diskPool [diskPoolName] locate
```

Parameters

Parameter	Description
`diskPool`	The alphanumberic identifier of the disk pool (including – and _) that you want to locate. Enclose the *diskPoolName* in square brackets ([ ]).

Start Enclosure Blink

Description

This command identifies an enclosure by turning on the indicator LED. (Use the `stop enclosure blink` command to turn off the indicator LED for the enclosure.)

Syntax

```
start enclosure [enclosureID] blink
```

Parameters

Parameter	Description
`enclosure`	Identifies the enclosure to blink. Enclosure ID values are `0` to `99`. You must put brackets ([ ]) around the enclosure ID value.

Start Input Output RAID Controller Module (IOC) Dump

Description

This command produces a dump of the IOC log of data transmissions between a host and a RAID controller module. RAID Controller Modules are identified as "0" or "1". Each RAID controller module can have up to four host channels, each channel is uniquely identified by a number from 1 through 4.

Syntax

```
start IOCLog [(0,1 | 0,2 | 1,1 | 1,2)] [overwrite=(TRUE | FALSE)]
```

Parameters

Parameter	Description
RAID controller module-channel identifiers	This parameter specifies the RAID controller module and host channel from which to produce the IOC dump. Valid RAID controller module identifiers are **0** or **1**. Enclose the RAID controller module identifier and host channel identifier in square brackets ([ ]). Valid RAID controller module identifier and host channel values are **0,1**, **0,2**, **1,1**, **1,2**, . If you do not specify a RAID controller module, the MD storage management software returns a syntax error.
overwrite	This parameter causes the new IOC log dump to overwrite an existing dump. To overwrite the existing dump, set this parameter to **TRUE**. The default value is **FALSE**.

NOTE: This command generates a debug log from the IOC of the selected RAID controller module and stores the data in a compressed format in a persistent memory buffer on the RAID controller module. You can retrieve the data from the debug log using the **save IOCLog** command. The RAID controller module returns an error for these conditions:

The RAID controller module platform and HIC do not support an IOC dump.

The specified RAID controller module has an outstanding IOC dump and the overwrite parameter is false.

The specified controller identifier or channel identifier are out of the valid range

Start iSCSI DHCP Refresh

Syntax

```
start controller [(0 | 1)] iscsiHostPort [(1 | 2 | 3 | 4)] dhcpRefresh
```

Parameters

Parameter	Description
`controller`	The identifier letter of the RAID controller module that has the iSCSI host ports. Valid RAID controller module identifier values are `0` or `1`, where, `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1.
`iscsi-host-port`	The identifier of the port on the storage array on which you want to refresh the DHCP parameters. Enclose the iSCSI host port name in square brackets ([ ]).

NOTE: This operation ends the iSCSI connections for the portal and brings down the portal temporarily.

Start Physical Disk Blink

Description

This command blinks a physical disk by turning on an indicator light on the physical disk. Run the **stop physicalDisk blink** command to turn off the indicator light on the physical disk.

Syntax

```
start physicalDisk [enclosureID,slotID] blink
```

Parameters

Parameter	Description
`physicalDisk`	The location of the physical disk that you want to find. For high-capacity expansion enclosures, specify the enclosure ID value and the slot ID value of the physical disk that you want to revive. Enclosure ID values are 0 to 99. Slot ID values are 0 to 31. Enclose the enclosure ID value and the slot ID value in square brackets ([ ]).

Start Physical Disk Channel Blink

Description

This command identifies the expansion enclosures that are connected to a specific physical disk port by turning on the indicator lights for the expansion enclosure that is connected to the physical disk port. Use the stop `physicalDiskChannel blink` command to turn off the indicator lights on the expansion enclosure.

Syntax

```
start physicalDiskChannel [(1 | 2)] blink
```

Parameters

Parameter	Description
`physicalDiskChannel`	The identifier number of the physical disk channel that you want to locate. Valid values for the identifier number are `1` or `2`. Enclose the identifier number in square brackets ([ ]).

Start Physical Disk Initialize

Description

This command starts physical disk initialization.

CAUTION: As soon as you enter this command, all user data is erased.

Syntax

```
start physicalDisk [enclosureID,slotID] initialize
```

Parameters

Parameter	Description
`physicalDisk`	Identifies the enclosure and slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put brackets ([ ]) around the enclosure ID value and slot ID value.

Start Physical Disk Reconstruction

Description

This command starts reconstructing a physical disk.

Syntax

```
start physicalDisk [enclosureID,slotID] reconstruct
```

Parameters

Parameter	Description
`physicalDisk`	Identifies the enclosure and slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot

Parameter	Description
	ID values are `0` to `31`. You must put brackets ([ ]) around the enclosure ID value and slot ID value.

Start Remote Replication Synchronization

Description

This command starts Remote Replication synchronization.

Syntax

```
start asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
synchronize [deleteRecoveryPointIfNecessary]
```

Parameter

Parameter	Description
asyncRemoteReplicationGroup	The name of the remote replication group for which you want to start synchronization. Enclose the remote replication group name in double quotation marks (" ") inside square brackets ([ ]").
deleteRecoveryPointIfNecessary	The parameter to delete the recovery point if the recoverable synchronization data has exceeded time threshold for recovery. Recovery point age is measured from the time the data was frozen on the primary storage array.

Start Remote Replication (Legacy) Synchronization

Description

NOTE: This command valid only with Fibre Channel storage arrays.

This command starts Remote Replication (legacy) synchronization.

Syntax

```
start remoteReplication primary
[”virtualDiskName”] synchronize
```

Parameters

Parameter	Description
`primary`	The name of the primary virtual disk for which you want to start synchronization. Enclose the primary virtual disk name in double quotation marks (“ ”) inside square brackets ([ ]).

Start Secure Physical Disk Erase

Description

This command erases all of the data from one or more self-encrypting disk (SED) physical disks so that they can be reused as SED physical disks. Run this command only when the SED physical disks are no longer part of a secure disk group, or when the security key is unknown.

NOTE: When performing a secure erase operation on all physical disks in an array that contains only SED physical disk type, the physical disks must be erased in two groups using the `start secureErase physicalDisks[]` command. For example, if there are seven (7) SED physical disks, the secure erase can be performed by executing:

```
start secureErase physicalDisks [0,0 0,1 0,2 0,3];
start secureErase physicalDisks [0,4 0,5 0,6];
```

Syntax

```
start secureErase (physicalDisk [enclosureID,slotID] |
physicalDisks [enclosureID1,slotID1 enclosureID2, slotID2...
enclosureIDn,slotIDn])
```

Parameters

Parameter	Description
`physicalDisk or physicalDisks`	Identifies the enclosure and slot where the physical disk resides. Enclosure ID values are `0` to `99`. Slot ID values are `0` to `31`. You must put brackets ([ ]) around the enclosure ID value and slot ID value.

Start Snapshot Rollback

Description

This command starts a rollback operation for one or more snapshot virtual disks. The content of the base virtual disk changes immediately to match the content of the selected snapshot virtual disk at that point in time. The base virtual disk immediately becomes available for read/write requests after the rollback operation has successfully completed. To stop a snapshot rollback operation, use the stop rollback virtual disk command. The repository virtual disk that is associated with the snapshot continues to track any new changes between the base virtual disk and the snapshot virtual disk that occurs after the rollback operation is completed.

Syntax

```
start rollback (virtualDisk [snapshotVirtualDiskName]
|

virtualDisks [snapshotVirtualDiskName1 … snapshotVirtualDiskNameN])
```

Parameters

Parameter	Description
`virtualDisk or` `virtualDisks`	Names the specific snapshot virtual disk or snapshot virtual disks for which you want to start a rollback operation. Enclose the snapshot virtual disk name in square brackets ([ ]). If the snapshot virtual disk name has special characters, you must also enclose the snapshot virtual disk name in double quotation marks (" ").

Start Snapshot Image Rollback

Description

This command starts a rollback operation for a set of snapshot images. The content of the base virtual disk changes immediately to match the point-in-time content of the selected snapshot image virtual disk. The base virtual disk immediately becomes available for read/write requests after the rollback operation has successfully completed. To stop a snapshot image rollback operation, use the `stop rollback snapImage` command.

The repository virtual disk that is associated with the snapshot image continues to track any new changes between the base virtual disk and the snapshot image virtual disk that occur after the rollback operation is completed.

NOTE: You cannot use this command for snapshot images involved in online virtual disk copy.

Syntax

```
start snapImage ["snapImageName"] rollback
```

Parameter

Parameter	Description
`snapImage`	The name of the snapshot image. The name of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group.

Parameter	Description
	Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]).

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group
- The identifier of the snapshot image

For example, if you want to start a rollback operation for snapshot image 12345 in a snapshot group that has the name **snapGroup1**, you would use this command:

```
start snapImage ["snapGroup1:12345"] rollback;
```

To start a rollback operation for the most recent snapshot image in a snapshot group that has the name **snapGroup1**, you would use this command:

```
start snapImage ["snapGroup1:newest"]rollback;
```

Start SSD Cache Performance Modeling

Description

This command starts performance modeling for the SSD cache. Performance modeling monitors and measures I/O activity for a period of time and estimates performance for various SSD cache sizes. Performance is estimated using two metrics: cache hit percentage and average response time. The performance modeling data is not available until you stop the operation using the `stop ssdCache performanceModeling` command.

Syntax

```
start ssdCache [ssdCacheName] performanceModeling
```

Parameter

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache that you want to locate. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Notes

Performance modeling ends and the performance modeling data is available when one of the following conditions occurs:

- Run the `stop ssdCache performanceModeling` command.

- Retrieve the performance modeling data using PowerVault MD Storage Manager.

Performance modeling ends, but no data is available when one of the following conditions occurs:

- You reboot the RAID controller module.
- You make any changes to the SSD cache configuration.
- The state of the SSD cache changes.

Start Storage Array Blink

Description

This command identifies a storage array by turning on the indicator LED for the storage array. Use the `stop storageArray blink` command to turn off the indicator LED for the storage array.

Syntax

```
start storageArray blink
```

Parameters

None.

Start Storage Array Configuration Database Diagnostic

Description

This command runs a consistency check against a configuration database.

Syntax

```
start storageArray configDbDiagnostic
[sourceLocation=(disk | onboard)]
[diagnosticType = (fileSystem | mirror | contentCheck)]
[controller[(0 | 1)]]
```

Parameters

Parameter	Description
`sourceLocation`	This parameter specifies the location of the database. • `disk` indicates that data comes directly from the database on the physical disk. • `onboard` indicates that data comes from the RPA memory location The default location is `disk`.
`diagnosticType`	The level of diagnostic testing that you want to run on the database. You can run one of these levels of testing:

Parameter	Description	
	fileSystem	This option checks the structural integrity of the database.
	replication	• When the `sourceLocation` parameter is set to `disk`, the peer RAID controller module initiates a block check. • When the `sourceLocation` parameter is set to `cache`, the peer RAID controller module initiates a record check.
	contentCheck	This option performs a deep analysis of each record's contents.
	The default value is `fileSystem`.	
`controller`	The RAID controller module that has the database on which you want to run the diagnostic tests. Valid RAID controller module identifiers are `0` or `1`, where `0` is the RAID controller module in slot 0, and `1` is the RAID controller module in slot 1. Enclose the RAID controller module identifier in square brackets ([ ]).	

Notes

This command runs a consistency check against a configuration database. All database records are checked. Error data is written to a file in the data folder on disk automatically. You do not need to specify an output file.

NOTE: Executing a consistency check with the `diagnosticType` parameter set to a value other than `filesystem`, and with the `sourceLocation` parameter set to `onboard`, can cause the operation to run for a long time. This can have adverse effects on host I/O processing. This operation should be done only under direction from the support organization.

Upon completion of the diagnostic test, the RAID controller firmware returns one of these results:

- Diagnosis completed without errors. No ZIP file created.
- Diagnosis completed with errors. Refer to the ZIP file created at: ...\Install_dir\data\FirmwareUpgradeReports\timestamp_buildNo.zip

If the diagnostic test detects an inconsistency in the configuration database, the RAID controller module firmware performs these actions:

- Returns a description of the inconsistency

- Saves a ZIP file containing raw binary data

The RAID controller module firmware saves the ZIP file to this location:...\Install_dir\data \FirmwareUpgradeReports\timestamp_buildNo.zip

You can use the binary data to help determine the cause of the problem, or you can send the file containing the binary data to a Technical Support representative.

To stop the database configuration diagnostic test, use the `stop storageArray configDbDiagnostic` command.

In addition, you can start the database configuration diagnostic test through the storage management software GUI; however, you cannot stop the database configuration diagnostic test through the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

Start Storage Array iSCSI Refresh

Description

This command initiates a refresh of the network address information for the iSNS server. If the DHCP server is marginal or unresponsive, the refresh operation can take from 2 to 3 minutes to complete.

NOTE: This command is for IPv4 only.

Syntax

```
start storageArray isnsServerRefresh
```

Parameters

None.

NOTE: If you used the `set storageArray isnsIPv4ConfigurationMethod` command to set the configuration but did not set the configuration to DHCP, running the `start storageArray isnsServerRefresh` returns an error.

Start Virtual Disk Initialization

Description

This command starts the formatting of a virtual disk in a storage array.

NOTE: Initializing a virtual disk is a long-running operation that you cannot stop.

Syntax

```
start virtualDisk [virtualDiskName] initialize
```

Parameters

Parameter	Description
`virtualDisk`	Name of the virtual disk for which to start the formatting. You must put brackets ([ ]) around the virtual disk name. If the virtual disk name has special characters, you must also put quotation marks (" ") around the virtual disk name.

Stop Configuration Database Diagnostic

Description

This command stops the diagnostic test to validate the configuration database in the RAID controller module firmware that was started by the `start storageArray configDbDiagnostic` command.

Syntax

```
Stop storageArray configDbDiagnostic
```

Parameters

None.

NOTE: The RAID controller module firmware returns a confirmation that the diagnostic test was cancelled. In addition, the database configuration diagnostic test can be started through the storage management software GUI but cannot be stopped through the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

Stop Consistency Group Snapshot Rollback

Description

This command stops a rollback operation to the member base virtual disks in a snapshot consistency group that was initiated by the `start rollback cgSnap` command.

NOTE: Canceling a consistency group snapshot rollback operation leaves the base virtual disk in an indeterminate state with potentially invalid or inconsistent data. The related consistency group snapshot virtual disk becomes disabled and unusable.

Syntax

```
stop cgSnapImage ["snapCGID:imageID"] | rollback [memberVirtualDiskSet
["memberVirtualDiskName1' ... "memberVirtualDiskNamen"] ]
```

Parameters

Parameter	Description
`consistencyGroupName`	The name of the consistency group for which you want to stop a rollback operation. Enclose the name of the consistency group in double quotation marks (" ") inside square brackets ([ ]). When the `memberVirtualDiskSet` parameter is not used, the rollback process applies to all member virtual disks of the consistency group.
`memberVirtualDiskSet`	The name of one or more member base virtual disks in a consistency group that you want to stop a rollback operation. Enclose each member base virtual disk name in double quotation marks (" ") inside parentheses (( )). You can enter more than one member virtual disk name. Enclose all of the member virtual disk names in one set of double quotation marks (" ") inside parentheses (( )). Separate each member virtual disk name with a white space.

Notes

To stop a roll back operation for an entire consistency group that has the name **congroup1**, you would use this command:

```
stop cgSnapImage ["congroup1"] rollback;
```

To start a roll back operation for base virtual disk members memVol1, memVol2, and memVol3 in a consistency group that has the name **congroup2**, you would use this command:

```
stop cgSnapImage ["congroup2"] rollback memberVirtualDiskset=("memVol1 memVol2
memVol3");
```

Stop Consistency Group Snapshot Virtual Disk

Description

This command stops the copy-on-write operation for creating a consistency group snapshot virtual disk. To restart the copy-on-write operation use the `resume cgSnapVirtualDisk` command.

NOTE: This command does not delete a consistency group snapshot virtual disk. To delete a consistency group snapshot virtual disk use the `delete cgSnapVirtualDisk` command.

Syntax

```
stop cgSnapVirtualDisk [snapVirtualDiskName]
```

Parameter

Parameter	Description
`cgSnapVirtualDisk`	The name of the consistency group snapshot virtual disk that you want to stop creating. Enclose the name of the consistency group snapshot virtual disk in square brackets ([ ]) The name of the consistency group snapshot virtual disk that you want to delete. Enclose the consistency group snapshot virtual disk name in square brackets ([ ]). If the consistency group snapshot virtual disk name has special characters, you also must enclose the snapshot virtual disk name in double quotation marks (" ").

Stop Disk Group Blink

Description

This command turns off the indicator LED on the physical disk that was turned on by the `start diskGroup blink` command.

Syntax

```
stop diskGroup blink
```

Parameters

None.

Stop Disk Pool Locate

Description

This command turns off the indicator lights on the physical disks that were turned on by the `start diskPool locate` command.

Syntax

```
stop diskPool locate
```

Stop Enclosure Blink

Description

This command turns off the indicator LED on the enclosure that was turned on by the `start enclosure blink` command.

Syntax

```
stop enclosure blink
```

Parameters

None.

Stop Pending Snapshot Images On Consistency Group

Description

This command stops all of the pending snapshot images that are to be created on a snapshot consistency group. If the snapshot consistency group does not have any pending snapshot images, the MD storage management software displays an error message and does not run the command.

Syntax

```
stop consistencyGroup [consistencyGroupName] pendingSnapImageCreation
```

Parameter

Parameter	Description
`consistencyGroupName`	The name of the consistency group for which you want to stop any pending snapshot operation. Enclose the name of the consistency group in square brackets ([ ]).

Stop Physical Disk Blink

Description

This command turns off the indicator light on the physical disk that was turned on by the `start physicalDisk blink` command.

Syntax

```
stop physicalDisk blink
```

Parameters

None.

Stop Physical Disk Channel Blink

Description

This command turns off the indicator lights on the expansion enclosures that were turned on by the `start physicalDiskChannel blink` command.

Syntax

```
stop physicalDiskChannel blink
```

Parameters

None.

Stop Physical Disk Replace

Description

This command stops a data copy operation that was started by either a **set physicalDisk** command or a **replace physicalDisk** command with the copyPhysicalDisk user input.

Syntax

```
stop physicalDisk replace ([enclosureID,drawerID,slotID] | <"wwID">)
```

Parameters

Parameter	Description
physicalDisk	The location of the physical disk that is being copied before you replace it. For high-capacity expansion enclosures, specify the enclosure ID value, the drawer ID value, and the slot ID value of the physical disk that you want to revive. For low-capacity expansion enclosures, specify the enclosure ID value and the slot ID value of the physical disk that you want to revive. Enclosure ID values are **0** to **99**. Drawer ID values are **0** to **4**. Slot ID values are **0** to **31**. Enclose the enclosure ID value, the drawer ID value, and the slot ID value in square brackets ([ ]).
physicalDisk	The World Wide Identifier (WWID) of the physical disk that you want to replace. Enclose the WWID in double quotation marks (" ") inside angle brackets (< >).

NOTE: The physicalDisk parameter supports both high-capacity expansion enclosures and low-capacity expansion enclosures. A high-capacity expansion enclosure has drawers that hold the physical disks. The drawers slide out of the expansion enclosure to provide access to the physical disk drives. A low-capacity expansion enclosure does not have drawers. For a high-capacity expansion enclosure, you must specify the identifier (ID) of the expansion enclosure, the ID of the drawer, and the ID of the slot in which a physical disk resides. For a low-capacity expansion enclosure, you need only specify the ID of the expansion enclosure and the ID of the slot in which a physical disk drive resides. For a low-capacity expansion enclosure, an alternative method for identifying a location for a physical disk is to specify the ID of the expansion enclosure, set the ID of the drawer to **0**, and specify the ID of the slot in which a physical disk drive resides.

Stop Snapshot Group Image

Description

This command cancels a pending snapshot image copy-on-write operation on a snapshot image group. If the snapshot image group does not have any pending snapshot images, this command returns an error message.

Syntax

```
stop (snapGroup [snapGroupName] | consistencyGroup [snapConsistencyGroupName])
pendingSnapImageCreation
```

Parameters

Parameter	Description
snapGroup	The alphanumeric identifier (including - and _) of the snapshot group that has the pending snapshot image copy-on-write operation that you want to stop. Enclose the snapshot group name in double quotation marks (" ").
consistencyGroup	The alphanumeric identifier (including - and _) of a snapshot consistency group that has the pending snapshot image copy-on-write operation that you want to stop. Enclose the snapshot consistency group name in double quotation marks (" ").

Stop Snapshot Group Pending Snapshot Images

Description

This command cancels all of the pending snapshot images that are to be created on a snapshot group. If the snapshot group does not have any pending snapshot images, the firmware displays an error message and does not run the command. You can run this command on a snapshot group or a snapshot consistency group.

Syntax

```
stop (snapGroup [snapGroupName] |
consistencyGroup [snapConsistencyGroupName])
pendingSnapImageCreation
```

Parameters

Parameter	Description
snapGroup	The alphanumeric identifier (including - and _) of the snapshot group for which you want to stop

Parameter	Description
	pending snapshot images. Enclose the snapshot group identifier in square brackets ([[]).
`consistencyGroup`	The alphanumeric identifier (including - and _) of the snapshot consistency group for which you want to stop pending snapshot images. Enclose the snapshot group identifier in square brackets ([[]).

Stop Snapshot (Legacy)

Description

This command stops a copy-on-write operation.

Syntax

```
stop snapshot (virtualDisk [virtualDiskName] |
virtualDisks [virtualDiskName1... virtualDiskNameN])
```

Parameters

Parameter	Description
`virtualDisk` or `virtualDisks`	The name of the specific virtual disk for which you want to stop a copy-on-write operation. You can enter more than one virtual disk name. Enclose the virtual disk names using one of these forms: • On a Windows command line: `\"virtualDiskName\"` • In a Windows script engine window: `["virtualDiskName"]` • On a Linux command line: `\"virtualDiskName\"` • In a Linux script engine window: `[\"virtualDiskName\"]`

Notes

Names can be any combination of alphanumeric characters, underscore (_), hyphen (-), and pound (#). Names can have a maximum of 30 characters.

One technique for naming the snapshot (legacy) virtual disk and the snapshot (legacy) repository virtual disk is to add a hyphenated suffix to the original base virtual disk name. The suffix distinguishes between the snapshot (legacy) virtual disk and the snapshot (legacy) repository virtual disk. For example, if you have a base virtual disk with a name of **Engineering Data**, the snapshot (legacy) virtual disk can have a name of **Engineering Data-S1**, and the snapshot (legacy) repository virtual disk can have a name of **EngineeringData- R1**.

If you do not choose a name for either the snapshot (legacy) virtual disk or the snapshot (legacy) repository virtual disk, the MD storage management software creates a default name by using the base virtual disk name. An example of the snapshot (legacy) virtual disk name that the RAID controller modules might create is, if the base virtual disk name is **aaa** and does not have a snapshot (legacy) virtual disk, the default snapshot (legacy) virtual disk name is **aaa-1**. If the base virtual disk already has *n*-1 number of snapshot (legacy) virtual disks, the default name is **aaa-***n*. An example of the snapshot (legacy) repository virtual disk name that the RAID controller module might create is, if the base virtual disk name is **aaa** and does not have a snapshot (legacy) repository virtual disk, the default snapshot (legacy) repository virtual disk name is **aaa- R1**. If the base virtual disk already has *n*-1 number of snapshot (legacy) repository virtual disks, the default name is **aaa-R***n*.

Stop Snapshot (Legacy) Rollback

Description

NOTE: With firmware version 7.83, the orderterms in the syntax is changed to be consistent with other commands. Replace the deprecated command syntax with the new command syntax.

This command stops a snapshot rollback operation that was initiated by the `start rollback virtual disk` command.

NOTE: Canceling a rollback operation leaves the base virtual disk in an indeterminate state with potentially invalid or inconsistent data. The related snapshot virtual disk becomes disabled and unusable.

Syntax

Deprecated

```
stop rollback (virtualDisk [snapshotVirtualDiskName] |
virtualDisks [snapshotVirtualDiskName1 ... snapshotVirtualDiskNameN])
```

New

```
stop (virtualDisk [snapshotVirtualDiskName ] |
virtualDisks [snapshotVirtualDiskName1 ... snapshotVirtualDiskNameN]) rollback
```

Parameters

Parameter	Description
`virtualDisk`	The name of the snapshot (legacy) virtual disk for which you want to stop a rollback operation. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters or numbers, you must enclose the virtual disk name in double quotation marks (" ") inside square brackets.
`virtualDisks`	The names of several snapshot (legacy) virtual disks for which you want to stop a rollback operation. Enter the namesvirtual disks using these rules: • Enclose all names in square brackets ([ ]). • Separate each name with a space.

Parameter	Description
	If the virtual disk names have special characters or numbers, enter the names using these rules: • Enclose all names in square brackets ([ ]). • Enclose each name in double quotation marks (" "). • Separate each name with a space.

Stop Snapshot Rollback

Description

This command stops a snapshot rollback operation that was initiated by the start rollback virtual disk command.

NOTE: Canceling a rollback operation leaves the base virtual disk in an indeterminate state with potentially invalid or inconsistent data. The related snapshot virtual disk becomes disabled and unusable.

Syntax

```
stop rollback virtualDisk [snapshotVirtualDiskName]
```

Parameters

Parameter	Description
`virtualDisk`	Names the specific snapshot virtual disk for which you want to resume a rollback operation. Enclose the snapshot virtual disk name in square brackets ([ ]). If the snapshot virtual disk name has special characters, you must also enclose the snapshot virtual disk name in double quotation marks (" ").

Stop Snapshot Image Rollback

Description

This command stops a snapshot image rollback operation that was initiated by the `start snapImage rollback` command.

NOTE: Canceling a snapshot image rollback operation leaves the base virtual disk in an indeterminate state with potentially invalid or inconsistent data. The related snapshot image virtual disk becomes disabled and unusable.

Syntax

```
stop snapImage [snapImageName] rollback
```

Parameter

Parameter	Description
`snapImage`	The name of the snapshot image for which you want to stop a rollback operation. The name of a snapshot image is comprised of two parts: • The name of the snapshot group. • An identifier for the snapshot image in the snapshot group. The identifier for the snapshot image can be one of these: • An integer value that is the sequence number of the snapshot in the snapshot group. • `NEWEST` - Use this option when you want to show the latest snapshot image created in the snapshot group. • `OLDEST` - Use this option when you want to show the earliest snapshot image created in the snapshot group. Enclose the snapshot image name in double quotation marks (" ") inside square brackets ([ ]).

Notes

The name of a snapshot image has two parts separated by a colon (:):

- The identifier of the snapshot group
- The identifier of the snapshot image

For example, if you want to stop a rollback operation for snapshot image 12345 in a snapshot group that has the name **snapGroup1**, you would use this command:

```
stop snapImage ["snapGroup1:12345"] rollback;
```

To stop a rollback operation for the most recent snapshot image in a snapshot group that has the name **snapGroup1**, you would use this command:

```
stop snapImage ["snapGroup1:newest"] rollback;
```

Stop Snapshot Virtual Disk

Description

This command stops a snapshot virtual disk operation.

Syntax

```
stop snapVirtualDisk ["snapVirtualDiskName"]
```

Parameter

Parameter	Description
`snapVirtualDisk`	The alphanumeric identifier (including - and _) of the snapshot virtual disk that you want to stop. Enclose the snapshot virtual disk identifier in double quotation marks (" ") inside square brackets ([ ]).

Stop SSD Cache Performance Modeling

Description

This command stops the performance modeling operation and displays the performance modeling data for the SSD cache. Before running this command, you must start the performance modeling operation with the `start ssdCache performanceModeling` command. Optionally, you can save the data to a file.

Syntax

`stop ssdCache [`*`ssdCacheName`*`] performanceModeling [file="`*`filename`*`"]`

Parameters

Parameter	Description
`ssdCache`	The alphanumeric identifier (including - and _) of the SSD cache that you want to locate. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.
`file`	The file path and the file name to which you want to save the performance modeling data. Enclose the file name in double quotation marks (" "). For example: **file="C:\Program Files\CLI\logs \performance.csv"**. The default name of the file that contains the performance modeling data is **readLinkStatus.csv**. You can use any file name, but you must use the **.csv** extension.

Notes

This command returns the performance modeling information similar to this example. The size of your monitor determines how the information wraps and will affect how the information appears.

```
SSD Cache Name: my_cache
Start time: 4/18/12 2:38:26 PM IST
Stop time: 4/18/12 2:38:45 PM IST
Duration : 00:00:19

SSD Cache Performance Modeling Data (Response Time):
```

		SSD Reads		HDD Reads		HDD Writes	
Cache Capacity	Overall Response Time	Avg. Response Time	% of I/Os	Avg. Response Time	% of I/Os	Avg. Response Time	% of I/Os
186 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
372 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
557 GB *	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
558 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
744 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
931 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
1117 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
1303 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
1489 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
1675 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %
1862 GB	0 ms	0 ms	0.0 %	0 ms	0.0 %	0 ms	0.0 %

* = Current SSD cache physical capacity.

SSD Cache Performance Modeling Data (Cache Hit %):

Cache Capacity	Cache Hit %
186 GB	0 %
372 GB	0 %
557 GB *	0 %
558 GB	0 %
744 GB	0 %
931 GB	0 %
1117 GB	0 %
1303 GB	0 %

```
1489 GB                                  0 %
1675 GB                                  0 %
1862 GB                                  0 %
```

* = Current SSD cache physical capacity.

Stop Storage Array Blink

Description

This command turns off the indicator lights on the storage array that were turned on by the `start storageArray blink` command.

Syntax

```
stop storageArray blink
```

Parameters

None.

Stop Storage Array Configuration Database Diagnostic

Description

This command stops the diagnostic test to validate the configuration database in the RAID controller module firmware that was started by the `start storageArray configDbDiagnostic` command.

Syntax

```
stop storageArray configDbDiagnostic
```

Parameters

None.

Notes

The RAID controller module firmware returns a confirmation that the diagnostic test was cancelled.

In addition, you can start the database configuration diagnostic test through the storage management software GUI; however, you cannot stop the database configuration diagnostic test through the storage management software GUI. If you want to stop a running diagnostic test, you must use the `stop storageArray configDbDiagnostic` command.

If you try to use the `stop storageArray configDbDiagnostic` command after validation of the storage array configuration is complete, you do not receive any message that the validation has finished. This behavior is expected.

Stop Storage Array iSCSI Session

Description

This command forces the termination of an iSCSI session.

Syntax

```
stop storageArray iscsiSession [sessionNumber]
```

Parameters

Parameter	Description
`sessioNumber`	The identifier number of the iSCSI session. Enclose the identifier number of the iSCSI session in square brackets ([ ]).

Stop Storage Array Physical Disk Firmware Download

Description

This command stops a firmware download to the physical disks in a storage array that was started with the `download storageArray physicalDiskFirmware` command. This command does not stop a firmware download that is already in progress to a physical disk; however, this command stops all firmware downloads to physical disks that are waiting for the download.

Syntax

```
stop storageArray physicalDiskFirmwareDownload
```

Parameters

None.

Stop Virtual Disk Copy

Description

This command stops a virtual disk copy operation.

Syntax

```
stop virtualDiskCopy target [targetName] [source [sourceName]]
```

Parameters

Parameter	Description
`target`	Name of the target virtual disk for which to stop a virtual disk copy operation. You must put brackets ([ ]) around the target virtual disk name. If the target virtual disk name has special characters, you must also put quotation marks (" ") around the target virtual disk name.
`source`	Name of the source virtual disk for which to stop a virtual disk copy operation. You must put brackets ([ ]) around the source virtual disk name. If the source virtual disk name has special characters, you must also put quotation marks (" ") around the source virtual disk name.

Suspend Remote Replication

Description

This command suspends a Remote Replication operation.

NOTE: This command is valid only with Fibre Channel storage arrays.

Syntax

```
suspend remoteReplication (primary
[primaryVirtualDiskName] |
primaries [primaryVirtualDiskName1 ...
primaryVirtualDiskNameN])
writeConsistency=(TRUE | FALSE)
```

Parameters

Parameter	Description
`primary or primaries`	The name of the virtual disk for which you want to suspend operation. Enclose the virtual disk name in square brackets ([ ]). If the virtual disk name has special characters, also enclose the virtual disk name in double quotation marks (" ").
`writeConsistency`	This parameter defines whether the virtual disks identified in this command are in a writeconsistency group or are separate. For the virtual disks in the same write-consistency group, set this parameter to `TRUE`. For the virtual disks that are separate, set this parameter to `FALSE`.

NOTE: If the `writeConsistency` parameter is set to `TRUE`, the virtual disks must be in a write-consistency group (or groups). This command suspends all write-consistency groups that contain the virtual disks. For example, if virtual disks A, B, and C are in a write-consistency group and they have remote counterparts A', B', and C', the command:

```
suspend remoteReplication virtualDisk ["A"]
writeConsistency=TRUE
```

suspends A-A', B-B', and C-C'. If you have a write-consistency group 1={A, B, C} and write-consistency group 2={D, E, F}, the command:

```
suspend remoteReplication virtualDisks ["A" "D"]
writeConsistency=TRUE
```

suspends both write-consistency groups.

Suspend Remote Replication Group

Description

This command suspends the synchronization of data on all replicated pairs at the remote replication group level. This `suspend` command helps to reduce any performance impact to the host application that might occur while any changed data on the local storage array is copied to the remote storage array.

Syntax

```
suspend  asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
```

Parameter

Parameter	Description
`asyncRemoteReplicationGroup`	The name of the remote replication group that you want to suspend. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets. If the remote replication group name consists only of numbers, such as "1002," you also must enclose the remote replication group name in double quotation marks (" ") inside square brackets.

Suspend SSD Cache

Description

This command stops caching for all virtual disks that are using the SSD cache. Use this command to temporarily stop caching using the SSD cache. While caching is stopped, host reads are serviced from the base virtual disks instead of the SSD cache.

Syntax

```
suspend ssdCache [ssdCacheName]
```

Parameter

Parameter	Description
ssdCache	The alphanumeric identifier (including - and _) of the SSD cache that you want to suspend. Enclose the identifier in square brackets ([ ]). If the SSD cache name contains special characters or consists only of numbers, you also must enclose the identifier in double quotation marks (" ") inside square brackets.

Notes

To restart caching, use the `resume ssdCache` command.

Test Remote Replication Group Connectivity

Description

This command tests possible communication problems between the local storage array and the remote the storage array associated with a remote replication group.

Syntax

```
diagnose  asyncRemoteReplicationGroup ["asyncRemoteReplicationGroupName"]
testID=(all | connectivity | latency | bandwidth | portConnections)
```

Parameters

Parameter	Description
asyncRemoteReplicationGroup	The name of an existing remote replication group that you want to test. Enclose the remote replication group name in square brackets ([ ]). If the remote replication group name has special characters, you also must enclose the remote replication group name in double quotation marks (" ").
testID	The identifier for the diagnostic test you want to run. The identifier and corresponding tests are as follows: • All—Performs all the tests associated with this command. • Connectivity—Verifies that the two RAID controller modules have a communication

Parameter	Description
	path. The connectivity test sends an inter-RAID controller module message between the storage arrays, and then validates that the corresponding remote replication group on the remote storage array exists. It also validates that the virtual disk members of the remote replication group on the remote system match the virtual disk members of the remote replication group on the local system. • `Latency`—Sends a SCSI test unit command to each virtual disk on the remote storage array associated with the remote replication group to test the minimum, average, and maximum latency. • `Bandwidth`—Sends two inter-RAID controller module messages to the remote storage array to test the minimum, average, and maximum bandwidth as well as the negotiated link speed of the port on the RAID controller module performing the test. • `Port connections`—Shows the port that is being used for replicating on the local storage array and the port that is receiving the replicated data on the remote storage array.

Test SNMP Trap Destination

Description

This command tests the connection and identification of the trap receiver for a specific Simple Network Management Protocol (SNMP) community. This command tests the trap destination by sending a trap message to the trap receiver.

Syntax

```
start snmpTrapDestination trapReceiverIP=ipAddress
communityName="communityName"
```

Parameters

Parameter	Description
`trapReceiverIP`	The IP address of the SNMP manager to which you want to send trap messages.
`communityName`	The name of the SNMP community for which you want to send trap messages.

Unregister SNMP Community

Description

This command deletes an existing Simple Network Management Protocol (SNMP) community that you previously created and registered using the `create snmpCommunity` command. Deleting an SNMP community effectively unregisters the community. Any trap destinations associated with that community are also removed.

Syntax

```
delete snmpCommunity communityName="snmpCommunityName"
```

Parameters

Parameter	Description
communityName	The name of the SNMP community that you want to delete. Enclose the SNMP community name in double quotation marks (" ").

Unregister SNMP Trap Destination

Description

This command deletes an existing Simple Network Management Protocol (SNMP) trap destination that you previously created and registered using the `create snmpTrapDestination` command.

Syntax

```
delete snmpTrapDestination trapReceiverIP=ipAddress
```

Parameters

Parameter	Description
trapReceiverIP	The IP address of the SNMP manager from which you want to stop sending trap messages.

Update SNMP Community

Description

This command creates a new name for an existing Simple Network Management Protocol (SNMP) community.

Syntax

```
set snmpCommunity communityName="snmpCommunityName"
newCommunityName="newSnmpCommunityName"
```

Parameters

Parameter	Description
communityName	The name of the existing SNMP community that you want to rename. Enclose the SNMP community name in double quotation marks (" ").
newCommunityName	The new name that you want to give to the SNMP community. Enclose the SNMP community name in double quotation marks (" ").

Update SNMP MIB II System Group Variables

Description

This command changes the system variables for the Simple Network Management Protocol (SNMP). The system variables are maintained in a Management Information Base II (MIB-II) database. The system variables are:

- Name of the system
- Name of the system contact
- Location of the system

Syntax

```
set snmpSystemVariables
sysName=newSystemName
sysContact=contactName
sysLocation=systemLocation
```

Parameters

Parameter	Description
sysName	The new name that you want to give to the SNMP system. Use standard SNMP and MIB conventions for the system name. Enclose the SNMP system name in square brackets([ ]).
sysContact	The name of contact person for the managed system, with information on how to contact this person. Enclose the SNMP contact name in square brackets([ ]).

Parameter	Description
sysLocation	The physical location of the system, such as "3rd flr". Enclose the SNMP system location in square brackets([ ]).

Update SNMP Trap Destination

Description

This command turns on or turns off sending authentication failure messages for an SNMP trap destination. Failure occurs when the SNMP agent received a message from an SNMP manager, but the message contained an invalid community name.

Syntax

```
set snmpTrapDestination trapReceiverIP=ipAddress
communityName="communityName"
[sendAuthenticationFailureTraps=(TRUE | FALSE)]
```

Parameters

Parameter	Description
trapReceiverIP	The IP address of the SNMP manager to which you want to send trap messages.
communityName	The name of the SNMP community for which you want to send trap messages.
sendAuthenticationFailureTraps	This parameter turns on or turns off sending authentication failure messages to an SNMP manager. To send authentication failure messages, set the parameter to **TRUE**. To prevent sending authentication failure messages, set the parameter to **FALSE**. The default is **TRUE**.

Validate Storage Array Security Key

Description

This command validates the security key for a storage array that has self-encrypting disk physical disks to make sure that the security key is not corrupt.

Syntax

```
validate storageArray securityKey
file="fileName"
passPhrase="passPhraseString"
```

Parameters

Parameter	Description
`file`	The file path and the file name to which you want to save the security key. For example, file="C:\Program Files\CLI\sup\seckey.slk" NOTE: You must add a file extension of **.slk** to the end of the file name.
`passPhrase`	A character string that encrypts the security key so that you can store the security key in an external file.

Additional Information

Your pass phrase must meet these criteria:

- The pass phrase must be between eight and 32 characters long.
- The pass phrase must contain at least one uppercase letter.
- The pass phrase must contain at least one lowercase letter.
- The pass phrase must contain at least one number.
- The pass phrase must contain at least one non-alphanumeric character, for example, < > @ +.

NOTE: If your pass phrase does not meet these criteria, you receive an error message.

13

Sample Script Files

This appendix provides sample scripts for configuring a storage array. These examples show how the script commands appear in a complete script file. You can copy these scripts and modify them to create a configuration unique to your storage array.

Create a script file in two ways:

- Use the `save storageArray configuration` command
- Write a script

By using the `save storageArray configuration` command, you can create a file to use to copy an existing configuration from one storage array to other storage arrays. You can also use this file to restore an existing configuration that has become corrupted. You can also copy an existing file to serve as a pattern from which you create a new script file by modifying portions of the original file. The default file extension is **.scr.**

Create a new script file using a text editor, such as Microsoft Notepad. The maximum line length is 256 characters. The command syntax must conform to the guidelines in Usage Guidelines and the rules in Command Formatting Rules. When creating a new script file, use any file name and extension that runs on the host operating system.

To run a script file from the command line, enter the following text:

```
client>smcli 123.45.67.89 -f scriptfile.scr;
```

Configuration Script Example 1

This example creates a new virtual disk using the `create virtualDisk` command in the free space of a disk group.

```
Show "Create RAID 5 Virtual Disk 7 on existing Disk Group 1";

//Create virtual disk on a disk group created by
the create virtual disk command

//Note: For disk groups that use all available
capacity, the last virtual disk on the disk group
is created using all remaining capacity by
omitting the capacity=virtualDiskCapacity parameter

create virtualDisk diskGroup=1 raidLevel=5
userLabel="7" owner=0 segmentSize=16 capacity=2GB;

show "Setting additional attributes for virtualDisk 7";
//Configuration settings that cannot be set during
virtualDisk creation
```

```
set virtualDisk["7"] mediaScanEnabled=false;
set virtualDisk["7"] consistencyCheckEnabled=false;
set virtualDisk["7"] modificationPriority=high;
```

This example shows blank lines between the lines beginning with **Show**, **Create**, **//Note**, and **create**. The blank lines are included in this example only for clarity. Each command is actually written on one line in the script file; however, the size of this page causes the command text to wrap. You might want to include blank lines in your script files to separate blocks of commands or make a comment more outstanding. You can do this by entering two forward slashes (//), which causes the script engine to treat the line as a comment.

The first line of text is the `show string` command. This command displays text bounded by quotation marks (" ") on a monitor screen when the script file runs. In this example, the text **Create RAID 5 Virtual Disk 7 on existing Disk Group 1** serves as a title describing the expected results of running this script file.

The line beginning **//Create** is a comment explaining that the purpose of this script file is to create a new virtual disk by using the `create virtualDisk` command on an existing disk group.

The line beginning **//Note:** is a comment in the script file explaining that the size of the last virtual disk created uses all of the available capacity because the *capacity* parameter is not used.

The command in this example creates a new virtual disk in disk group 1. The virtual disk has a redundant array of independent disks (RAID) level of 5. The virtual disk name (user label) is 7. (Note the quotation marks around the 7. The quotation marks indicate that the information in the marks is a label.) The new virtual disk is assigned to the RAID controller module in slot 0 in the RAID enclosure. The segment size is set to 16.

The following syntax is the general form of the command:

```
create virtualDisk diskGroup=diskGroupNumber
userLabel="virtualDiskName" [freeCapacityArea=
freeCapacityIndexNumber] [capacity=
virtualDiskCapacity | owner=(0 | 1) | segmentSize=
segmentSizeValue]
[enclosureLossProtect=(TRUE | FALSE)]
```

The general form of the command shows the optional parameters in a different sequence than the optional parameters in the example. You can enter optional parameters in any sequence. You must enter the required parameters in the sequence shown in the command descriptions.

The line **show "Setting additional attributes for virtual disk 7"** is another example of using the `show string` command. This command is placed here to tell you that the `create virtualDisk` command ran successfully. In addition, properties that could not be set by the `create virtualDisk` command are now set.

The `set virtualDisk` command parameters are shown on separate lines. You do not need to use separate lines for each parameter. You can enter more than one parameter with the `set virtualDisk` command by leaving a space between the parameters.

By using separate lines, however, you can more clearly see what parameters you are setting and the values to which you are setting the parameters. Blocking the parameters in this manner makes it easier to edit the file or copy specific parameter settings for use in another script file.

Configuration Script Example 2

This example creates a new virtual disk using the `create virtualDisk` command with user-defined physical disks in the storage array.

```
Show "Create RAID 5 Virtual Disk 2 on existing Disk Group 2";

//This command creates the disk group and the
initial virtual disk on that group.

//Note: For disk groups that use all available
capacity, the last virtual disk on the group is
created using all remaining capacity by omitting
the capacity=virtualDisk creation parameter

create virtualDisk raidLevel=5 userLabel="2"
physicalDisks=[0,1 0,6 1,7 1,3 2,3 2,6] owner=1
segmentSize=16 capacity=2GB;

show "Setting additional attributes for virtual
disk 7";
//Configuration settings that cannot be set during
virtual disk creation
set virtualDisk["7"] mediaScanEnabled=false;
set virtualDisk["7"] consistencyCheckEnabled=false;
set virtualDisk["7"] modificationPriority=high;
```

The command in this example, like the `create virtualDisk` command in the previous example, creates a new virtual disk. The significant difference between these two examples is that this example shows how you can define specific physical disks to include in the virtual disk. To find out what physical disks are available in a storage array, run the `show storageArray profile` command.

The following syntax is the general form of the `create virtualDisk` command shown in the previous example:

```
create virtualDisk raidLevel=(0 | 1 | 5 | 6)
userLabel="virtualDiskName" physicalDisks=
(enclosureID0,slotID0... enclosureIDn,slotIDn)
[capacity=virtualDiskCapacity | owner=(0 | 1) |
segmentSize=segmentSizeValue]
[enclosureLossProtect=(TRUE | FALSE)]
```